

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-143661

(43)Date of publication of application : 28.05.1999

(51)Int.CI.

G06F 3/12  
B41J 29/38

(21)Application number : 09-313178

(71)Applicant : CASIO COMPUT CO LTD

(22)Date of filing : 14.11.1997

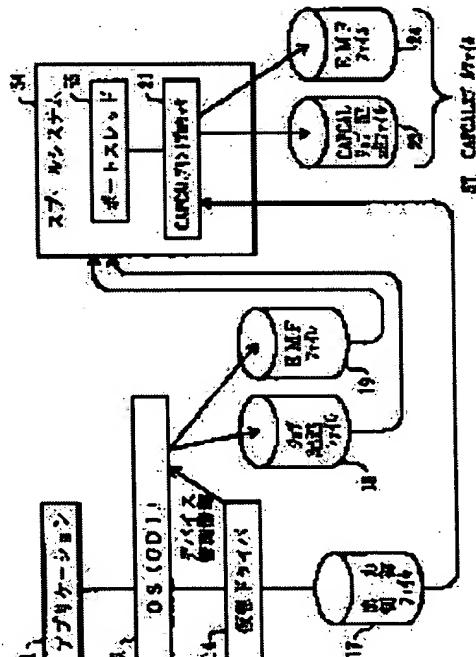
(72)Inventor : TSUTSUMI MASAKI  
WATANABE TAKAYASU  
TAMURA KOJI  
SHINTO YUKIHIRO  
HATANO EIJI  
AMARI TADAYOSHI

## (54) PRINTING SYSTEM

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a printing system which can deal with a document generated with a general-purpose application on a network.

**SOLUTION:** A virtual driver 14 is provided and GDI 2 spools a job described file 18 and an EMF file 19 and an output control file 17 is spooled by the virtual driver 14. Thus, the systems of the job described file 18 and the EMF file 19 are matched with the standard system of Windows. Document distribution processing, a dispersion/printing processing, a distribution destination automatic detection processing and the notifying processing of a printer error and printing completion are executed on all applications operating on the Windows containing general-purpose package software without changing the application interface of the windows.



## LEGAL STATUS

[Date of request for examination] 01.12.2000

[Date of sending the examiner's decision of 26.11.2002]

[rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-143661

(43)公開日 平成11年(1999)5月28日

(51)Int.Cl.<sup>6</sup>

G 0 6 F 3/12  
B 4 1 J 29/38

識別記号

F I

G 0 6 F 3/12  
B 4 1 J 29/38

C  
Z

審査請求 未請求 請求項の数 7 O L (全 30 頁)

(21)出願番号 特願平9-313178

(22)出願日 平成9年(1997)11月14日

(71)出願人 000001443

カシオ計算機株式会社  
東京都渋谷区本町1丁目6番2号

(72)発明者 簡見 勝紀

東京都羽村市栄町3丁目2番1号 カシオ  
計算機株式会社羽村技術センター内

(72)発明者 渡辺 隆保

東京都羽村市栄町3丁目2番1号 カシオ  
計算機株式会社羽村技術センター内

(72)発明者 田村 恒治

東京都羽村市栄町3丁目2番1号 カシオ  
計算機株式会社羽村技術センター内

(74)代理人 弁理士 大曾 義之

最終頁に続く

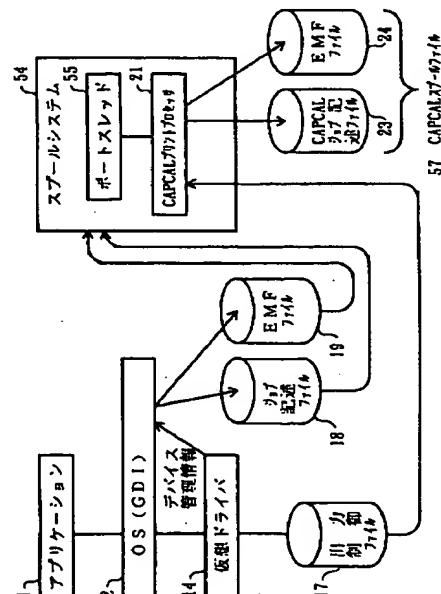
(54)【発明の名称】 印刷システム

(57)【要約】

【課題】 汎用アプリケーションで作成した文書をネットワーク上で取り扱うことが可能な印刷システムを提供する。

【解決手段】 仮想ドライバ14を設け、ジョブ記述ファイル18及びEMFファイル19を、GDI2によりスプールさせるとともに、出力制御ファイル17を仮想ドライバ14によりスプールさせることにより、ジョブ記述ファイル18及びEMFファイル19の形式を、Windows標準の形式と一致させ、Windowsのアプリケーションインターフェースを一切変更せずに、汎用パッケージソフトを含むWindows上で動作する全てのアプリケーションについて、ドキュメント配布処理、分散印刷処理、配布先自動検出処理、プリンタエラー及び印刷完了の通知処理等を行う。

仮想ドライバの動作を説明するブロック図



1

2

## 【特許請求の範囲】

【請求項1】 描画データを生成する描画データ生成手段と、前記描画データ生成手段を管理するオペレーティングシステムと、前記描画データによるスプールファイルの生成を、前記オペレーティングシステムに対して指示する指示手段と、前記描画データの印刷条件を設定する印刷条件設定手段と、前記印刷条件に基づいて、前記描画データの出力制御情報を生成する出力制御情報生成手段と、前記出力制御情報を格納する出力制御情報格納手段とを備えることを特徴とする印刷システム。

【請求項2】 前記出力制御情報生成手段により生成された出力制御情報に基づいて、前記描画データに対応したプリンタ管理ファイルを生成するプロセッサをさらに備えることを特徴とする請求項1に記載の印刷システム。

【請求項3】 前記プリンタ管理ファイルの内容及び前記スプールファイルの内容を受信する受信手段と、配布先名と出力装置名との対応関係を記述した配布先定義テーブルと、前記プリンタ管理ファイルの内容から印刷ジョブの検索条件を取得する検索条件取得手段と、前記検索条件に基づいて、前記印刷ジョブから文字列を抽出する文字列抽出手段と、前記文字列抽出手段により抽出された文字列と、配布先定義テーブルに記述された配布先名とを比較する比較手段と、前記配布先名と一致する文字列が抽出された印刷ジョブを、前記配布先名に対応する出力装置に出力する出力手段とを備えることを特徴とする請求項1または2に記載の印刷システム。

【請求項4】 前記印刷条件設定手段は、仮想的な用紙を表示する仮想用紙表示手段と、前記仮想的な用紙を分割する分割手段と、前記分割された矩形領域を指定する分割領域指定手段とを備え、前記文字列抽出手段は、前記分割領域指定手段により指定された矩形領域に対応する印刷ジョブの範囲から、文字列を抽出することを特徴とする請求項3に記載の印刷システム。

【請求項5】 前記プリンタ管理ファイルの内容及び前記スプールファイルの内容を受信する受信手段と、配布先名と出力装置名との対応関係を記述した配布先定義テーブルと、グループ名と前記配布先名へのリンク情報を記述したグループ定義テーブルと、前記プリンタ管理ファイルの内容から配布先情報を取得

する配布先情報取得手段と、前記配布先情報の指定配布モードがグループ指定の場合、前記配布先情報で示されるグループ名とリンクした配布先名を前記グループ定義テーブルから検索し、前記配布先名に対応する出力装置に印刷ジョブを出力する第1の出力手段と、

前記配布先情報の指定配布モードが論理名指定の場合、前記配布先情報で示される論理名と一致する配布先名を前記配布先定義テーブルから検索し、前記配布先名に対応する出力装置に印刷ジョブを出力する第2の出力手段と、

前記配布先情報の指定配布モードが物理名指定の場合、前記配布先情報で示される物理名と一致する出力装置に印刷ジョブを出力する第3の出力手段とを備えることを特徴とする請求項1または2に記載の印刷システム。

【請求項6】 スプールファイルの送信結果を印刷要求元情報とともに記録する送信結果記録手段と、前記スプールファイルの受信結果を前記印刷要求元情報とともに記録する受信結果記録手段と、

前記スプールファイルの出力結果を前記印刷要求元情報とともに記録する出力結果記録手段と、前記スプールファイルの印刷結果を前記印刷要求元情報とともに記録する印刷結果記録手段と、前記スプールファイルの出力先が自己的サーバの管轄下のプリンタである場合、前記印刷結果記録手段に記録された印刷結果を探査し、前記スプールファイルの出力先が他のサーバの管轄下のプリンタである場合、前記送信結果記録手段に記録された送信結果を探査する探索手段とをさらに備えることを特徴とする請求項1～5のいずれか1項に記載の印刷システム。

【請求項7】 描画データを生成する機能と、オペレーティングシステムを実現する機能と、前記描画データによるスプールファイルを前記オペレーティングシステムに生成させる機能と、前記描画データの印刷条件を設定する機能と、前記印刷条件に基づいて、前記描画データの出力制御情報を生成する機能と、前記出力制御情報を格納する機能とをコンピュータに実行させるプログラムを格納したコンピュータ読み取り可能な記憶媒体。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は印刷システムに関し、特に、クライアントサーバ印刷システムに適用して好適なものである。

## 【0002】

【従来の技術】 近年、オフィス業務や基幹業務向けコンピュータのオペレーティングシステムとして、マイクロソフト社のWindowsの普及が目覚ましい。

【0003】 図26は、オペレーティングシステムとし

て、Windowsを用いた場合の従来の印刷方法を説明するブロック図である。図26において、ユーザが、アプリケーション201にプリンタドライバ203に対しての印刷を指定した場合、アプリケーション201は、GDI（グラフィカルデバイスインターフェース関数）202に対して印刷命令を実行する。GDI202は、アプリケーション201からの印刷命令が実行されると、プリンタドライバ203から必要な情報を獲得し、スプールファイル（EMFファイルとジョブ記述ファイル）204を作成する。Windowsのスプールシステム205は、指定のポートが印刷可能状態であることを確認すると、デスプール処理を開始する。このデスプール処理により、ポートスレッド206を通じてプリントプロセッサ207に印刷制御が渡る。そして、ローカルプリントプロバイダ208を通じてポートモニタ209に印刷データが渡り、パラレルポートドライバ210を通じて印刷装置211に印刷データが出力される。

【0004】ここで、オペレーティングシステムとして、Windowsを用いた場合、Windowsはパソコン指向のオペレーティングシステムであることから、基幹業務における大量印刷には不向きな点がある。例えば、1回の印刷アクションで、同時に複数のプリンタに印刷することができない。このため、資料の配布等を行うとき、繰り返し印刷起動をかけ、印刷物を手作業で仕分けし、さらに郵便等で配布する手間がかかる。また、昨今のメールの普及により、ワードプロセッサなどにより作成された少量かつ固定的ドキュメントは、メールで代替えすることができるが、アプリケーション稼働による集計データ等の管理帳票の印刷や大量のドキュメントの印刷などは、メールでは代替不可能である。さらに、大量の印刷を行う時、複数のプリンタに分散して印刷することができず、開始頁と終了頁と出力先プリンタとを繰り返し指定する必要がある。

【0005】これらの問題を解決するため、従来では、基幹業務帳票印刷用の専用のAPI（アプリケーション・インターフェース関数）を用意し、そのAPIを用いて、Windowsなどのオペレーティングシステムの有する印刷制御の不備を補う方法があった。

【0006】図27は、従来の分散印刷方法を説明するブロック図である。図27において、アプリケーション221は、専用の印刷制御関数として、API223を呼び出す。ここで、API223には、プリンタ名称、開始頁及び終了頁の組を複数受け取ることができるインターフェースが設られ、API223にこのインターフェースを設けることにより、分散印刷を可能としている。API223が、パラメータ222として、プリンタ名称、開始頁及び終了頁の組を複数受け取るとともに、描画データをアプリケーション221から受け取ると、受け取ったパラメータ222と描画データとを、オ

ペレーティングシステム224及びファイルドライバ225を経由してスプールファイル226に格納する。

【0007】一方、スプールライタ227は、ある決められたディレクトリ（スプールディレクトリ）をポーリングしており、完成したスプールファイル226を検知すると、スプールファイル226の読み出しを開始する。そして、スプールファイル226内に格納されたパラメータ222を取り出すとともに、各ページの描画データをスプールファイル226から順次読み出す。その後、GDI228が呼び出され、パラメータ222の情報に従って、描画データがプリンタドライバ229を介して指定のプリンタ230へ送られる。この結果、指定の頁の印刷を指定のプリンタ230で行うことが可能となり、プリンタ名称、開始頁及び終了頁の組を複数指定しておくことにより、複数のプリンタを用いた分散印刷が可能となる。

【0008】なお、GDI228は、オペレーティングシステム224と密接に結びついたグラフィカルデバイス描画用の関数群である。

【0009】

【発明が解決しようとする課題】しかしながら、従来のAPI223を提供する手法では、API223を呼び出すことが可能な専用のアプリケーション221だけにしか、適用することができなかった。このため、ワードプロセッサなどの汎用パッケージアプリケーションには、API223を呼び出す機能がないため、ワードプロセッサなどの汎用パッケージアプリケーションには適用することができないという問題があった。また、提供されたAPI223を用いて印刷用のアプリケーション221を構築しなければならないという問題もあった。

【0010】さらに、従来の印刷方法では、オペレーティングシステム224の印刷制御の不備からくる問題点があり、API223を提供する方法では、これらの問題点を解決することができなかった。例えば、プリンタが接続されたコンピュータ上の画面には、印刷が完了したことやエラーが発生したことを知らせる通知があるが、通信ネットワークを介して印刷実行を指示したクライアントマシンでは、これらのことことが判らない。また、プリンタエラーで停止中のジョブに対し、頁を指定し直した再印刷や出力プリンタを変更した再印刷ができない。さらに、WAN（広域ネットワーク）で接続されたコンピュータ上のプリンタを直接指定して印刷することができない。

【0011】また、オペレーティングシステム224の出力装置・アプリケーションへの依存性に関する問題点として、以下のものがあった。まず、プリンタ機種やプリンタメーカーを特定して印刷機能が提供されており、ネットワーク環境の多様なプリンタを有効に使うためには、多大な手間がかかった。このため、特定出力装置（プリンタなど）のスペックに基づきアプリケーション

により作成したドキュメントは、出力装置の物理仕様に依存したものとなり、可搬性がなかった。この結果、物理仕様の異なる出力装置に出力すると、期待する結果を得られない場合があった。

【0012】次に、システム上の全アプリケーションで出力したドキュメントを対象とした場合、アプリケーションで出力後は、何もすることができなかつた。例えば、出力先の追加・変更、頁単位での操作、出力順の操作など様々なサービスが不可能であった。

【0013】また、オペレーティングシステム224の配布印刷方式に関する問題点として、以下のものがあった。まず、印刷出力（ドキュメント）を配布する場合、配布先のプリンタを直接指示するものであり、システム環境（配布先に対するプリンタ名称）を常に意識する必要があった。また、配布対象となるプリンタの名称を全て指定したり、ドキュメント毎に配布先を指定したりするすることは、ユーザにとって負担になっていた。

【0014】次に、ドキュメント内容から配布先を決定する場合、配布先を決定するためのドキュメントの領域を指定する必要があるが、指定した領域の文字列全てが配布先候補として扱われる所以、確実に配布先を指示するためには、指定した領域で使用できる文字列に制限が発生する。このため、ある配布先を一時的に対象外とする場合、ドキュメントから文字列を削除するなどの複雑な操作や後の修復の手間などが必要となり、誤操作の要因となる場合があった。さらに、自由度が高いため実現方法が複雑で性能などに課題が発生する。

【0015】また、オペレーティングシステム224の領域指定方式に関する問題点として、以下のものがあった。何からの目的で印刷ドキュメントの領域を指定することが必要な場合、各アプリケーションにその機能を組み込む必要があるが、各ドキュメントの形状（用紙サイズ、置き方）に依存した設定方式であるため、ドキュメント毎に設定を行う必要があった。さらに、一度設定しても、ドキュメントの形状を変更するたびに再設定するという手間があった。

【0016】また、オペレーティングシステム224のドキュメント出力監視制御方式に関する問題点として、以下のものがあった。印刷処理（プリンタ出力）の管理であり、既存のシステムでは、プリンタを主体とした監視機能は提供されていたが、各ユーザが投入した印刷処理を主体とした監視が不可能であり、ユーザは、自分が投入した印刷処理全てを、簡単に監視することが困難であった。

【0017】そこで、本発明の第1の目的は、汎用アプリケーションで作成した文書をネットワーク上で取り扱うことが可能な印刷システムを提供することである。また、本発明の第2の目的は、汎用アプリケーションで作成した文書の配布印刷を容易に行うことが可能な印刷システムを提供することである。

【0018】また、本発明の第3の目的は、ネットワークに接続されているプリンタの動作状態をクライアント側で監視することが可能な印刷システムを提供することである。

【0019】

【課題を解決するための手段】上述した課題を解決するために、本発明によれば、描画データを生成する描画データ生成手段と、前記描画データ生成手段を管理するオペレーティングシステムと、前記描画データに基づくスプールファイルの生成を、前記オペレーティングシステムに対して指示する指示手段と、前記描画データの印刷条件を設定する印刷条件設定手段と、前記印刷条件に基づいて、前記描画データの出力制御情報を生成する出力制御情報生成手段と、前記出力制御情報を格納する出力制御情報格納手段とを備えている。

【0020】このことにより、描画データ生成手段により生成された描画データを、その描画データ生成手段を管理するオペレーティングシステムによりスプールさせることができとなることから、そのオペレーティングシステム上で動作するアプリケーションならば、どのようなアプリケーションも、描画データ生成手段として使用することが可能となる。また、出力制御情報を格納する出力制御情報格納手段を設けることにより、出力制御情報を描画データに基づくスプールファイルと別々に格納することが可能となることから、オペレーティングシステムに備わっていなかった出力制御機能を、後から自由に追加することが可能となる。この結果、従来のオペレーティングシステムでは容易に実現することができなかつたネットワーク環境でのドキュメント配布処理、分散印刷処理、配布先自動検出処理、またはプリンタエラー及び印刷完了の通知処理等を、汎用アプリケーションで作成した文書に対して行うことが可能となる。

【0021】また、本発明の一態様によれば、前記出力制御情報生成手段により生成された出力制御情報に基づいて、前記描画データに対応したプリンタ管理ファイルを生成するプロセッサをさらに備えている。

【0022】このことにより、オペレーティングシステムと独立した独自の形式でプリンタ管理ファイルの処理や操作を行うことが可能となり、ドキュメントの印刷処理を多種多様な形態で柔軟に行うことが可能となる。

【0023】また、本発明の一態様によれば、配布先名と出力装置名との対応関係を記述した配布先定義テーブルと、受信したプリンタ管理ファイルの内容から印刷ジョブの検索条件を取得する検索条件取得手段と、前記検索条件に基づいて、前記印刷ジョブから文字列を抽出する文字列抽出手段と、前記文字列抽出手段により抽出された文字列と、配布先定義テーブルに記述された配布先名とを比較する比較手段と、前記配布先名と一致する文字列が抽出された印刷ジョブを、前記配布先名に対応する出力装置に出力する出力手段とを備えている。

【0024】のことにより、クライアントから送られた検索条件に基づいて、ドキュメントに含まれる予め指定された文字列を自動的に判別することが可能となり、印刷ジョブから配布先を自動的に検出することが可能となるとともに、配布先定義テーブルを参照することにより、自動的に検出された配布先に基づいて、出力装置を特定することが可能となる。この結果、印刷ドキュメントの配布先を指定する手間を省くことが可能となり、配布先に対応するプリンタ名称を意識することなく、1回の印刷指示を行うだけで検索条件に該当する全ての配布先へ自動的に印刷を行うことが可能となる。

【0025】また、本発明の一態様によれば、仮想的な用紙を表示する仮想用紙表示手段と、前記仮想的な用紙を分割する分割手段と、前記分割された矩形領域を指定する分割領域指定手段とを備え、前記分割領域指定手段により指定された矩形領域に対応する印刷ジョブの範囲から、文字列を抽出するようにしている。

【0026】のことにより、検索領域を用紙の上下左右や真ん中などの方向で指定することが可能となることから、配布先の検索などのため、印刷ドキュメントの領域を指定することが必要な場合、仮想的な用紙への1回の領域設定により、各印刷ドキュメントの形状（用紙サイズ、置き方）に依存することなく、配布先の検索領域を適切に設定することが可能となる。

【0027】また、本発明の一態様によれば、配布先名と出力装置名との対応関係を記述した配布先定義テーブルと、グループ名と前記配布先名へのリンク情報を記述したグループ定義テーブルと、受信したプリンタ管理ファイルの内容から配布先情報を取得する配布先情報取得手段と、前記配布先情報の指定配布モードがグループ指定の場合、前記配布先情報で示されるグループ名とリンクした配布先名を前記グループ定義テーブルから検索し、前記配布先情報の指定配布モードが論理名指定の場合、前記配布先情報で示される論理名と一致する配布先名を前記配布先定義テーブルから検索し、前記配布先情報の指定配布モードが物理名指定の場合、前記配布先情報で示される物理名と一致する出力装置に印刷ジョブを出力するようにしている。

【0028】のことにより、N台分の配布先名と出力装置との対応関係を登録したり、1つの論理的な配布先に複数の出力装置を定義することが可能となり、配布先を1つ指定するだけで、その配布先に対応する複数の出力装置を自動的に決定することが可能となることから、配布先を物理的な出力装置ではなく、論理的な名称で指定することが可能となるとともに、1回の指定により複数の出力装置にドキュメントを配布することが可能となる。このため、希望する配布先を印刷実行時に複数選択したり、印刷ドキュメント毎に配布先を指定したり、配布対象であるプリンタの名称を全て指定したりする手間を省くことが可能となり、配布先に対応するプリンタ名

を意識することなく、1回の印刷指示で条件に該当する全ての配布先へ自動的に印刷を行うことが可能となる。

【0029】また、本発明の一態様によれば、スプールファイルの送信結果を印刷要求元情報とともに記録する送信結果記録手段と、前記スプールファイルの受信結果を前記印刷要求元情報とともに記録する受信結果記録手段と、前記スプールファイルの出力結果を前記印刷要求元情報とともに記録する出力結果記録手段と、前記スプールファイルの印刷結果を前記印刷要求元情報とともに記録する印刷結果記録手段と、前記スプールファイルの出力先が自己のサーバの管轄下のプリンタである場合、前記印刷結果記録手段に記録された印刷結果を探索し、前記スプールファイルの出力先が他のサーバの管轄下のプリンタである場合、前記送信結果記録手段に記録された送信結果を探索する探索手段とを備えていることにより、ネットワーク上でのスプールファイルの転送が行われた場合においても、そのスプールファイルの送信結果、受信結果、出力結果及び印刷結果が、どの印刷要求元に起因して得られたのかを容易に判定することが可能となり、送信結果記録手段、受信結果記録手段、出力結果記録手段及び印刷結果記録手段の内容を調べることにより、自分の印刷要求がどのサーバのプリンタでどのように処理されたのかを認識することが可能となる。この結果、クライアントは、プリンタがどのサーバで管理されているのかを意識することなく、印刷要求を行うことが可能となるとともに、いずれのサーバにおいて印刷が行われた場合においても、印刷状況を正確に把握することができる。

【0030】

【発明の実施の形態】以下、本発明の一実施例に係わる印刷システムについて図面を参考しながら説明する。なお、以下の実施例では、Windowsによるオペレーティングシステム下における出力装置運用支援（以下CAPCALと称する）制御方式について説明する。また、このCAPCAL制御方式では、システム上の全てのアプリケーションの出力ドキュメントを処理対象とすることができる。

【0031】図1は、本発明の一実施例に係わる印刷システムの構成を示すブロック図である。図1において、クライアント300とサーバ301とが通信ネットワーク302、303を介して接続され、サーバ301はプリンタ9と接続されている。ここで、通信ネットワーク302は、例えば、LAN（ローカルエリアネットワーク）、通信ネットワーク303は、例えば、LAN、WAN（ワイドエリアネットワーク）、インターネットである。クライアント300には、アプリケーション1、GDI2、Windows印刷制御部3及びクライアント制御部4が設けられ、サーバ301には、Windows印刷制御部5、サーバ制御部6、GDI7及びプリ

ンタ出力制御部8が設けられている。そして、Windowsによる印刷処理を行う場合、Windows印刷制御部3、5による処理が行われ、CAPCAL制御を行う場合、クライアント制御部4、サーバ制御部6及びプリンタ出力制御部8による処理が行われる。なお、CAPCAL制御を選択する場合、CAPCAL制御用に設けられたCAPCALドライバをアプリケーション1で指定してGDI2を呼び出す。

【0032】図2は、図1のクライアント制御部4の構成を説明するブロック図である。図2において、クライアント制御部4には、CAPCALドライバ14、CAPCALドライバプロパティ制御スレッド15、出力制御ファイル17、ジョブ記述ファイル18、EMFファイル19、24、CAPCALクライアント定義ファイル20、CAPCALプリントプロセッサ21、CAPCALジョブ記述ファイル23、CAPCAL転送先定義ファイル25、CAPCALクライアントメインプロセス26及びCAPCALジョブ転送スレッド27が設けられている。また、Windows印刷制御部3には、スプーラプロセス11、ローカルプリントプロバイダ12及びプリントプロセッサ13が設けられている。ここで、クライアント制御部4は、Windowsによるオペレーティングシステム上で稼働するアプリケーション1が描画した印刷データをCAPCALの処理環境に取り込み、サーバ301へ転送するまでの処理を制御する。

【0033】アプリケーション1が描画した印刷データをCAPCAL環境で処理する場合、アプリケーション1は、仮想ドライバとしてのCAPCALドライバ14を指定し、GDI2を呼び出す。GDI2は、CAPCALドライバ14から描画機能を取得し、アプリケーション1の要求した描画プリミティブをCAPCALドライバ14が有する描画機能に変換して、システムスプールキュー16にスプールする。このシステムスプールキュー16には、印刷要求したユーザ名や、アプリケーション名などを格納したジョブ記述ファイル18と、描画情報をフォーマット化して格納したEMFファイル19が作成される。EMFとは、Enhanced Meta-Fileの略で、Windowsにおける描画データの論理的な格納形式である。なお、ジョブ記述ファイル18及びEMFファイル19の形式は、Windows標準の形式と完全に同一である。このため、ジョブ記述ファイル18及びEMFファイル19は、Windowsに備わっているデフォルト機能により生成させることができる。

【0034】また、CAPCALドライバ14は、CAPCAL特有の出力制御情報を出力制御ファイル17として、システムスプールキュー16に書き込む。なお、この出力制御情報は、CAPCALドライバ14側からダイアログ画面を開き、配布や分散などの印刷条件をオ

ペレータに指定させることにより、設定することができる。ここで、オペレータにより入力された出力制御情報を文書ごとに記憶し、次回にその文書が印刷される時は、前回入力されたその文書の出力制御情報をデフォルトとして使用するようにしても良い。

【0035】アプリケーション1の全描画データのスプールが完了すると、CAPCALプリントプロセッサ21が呼び出される。このCAPCALプリントプロセッサ21の呼び出しは、アプリケーション1でCAPCALドライバ14を指定することにより行われる。なお、アプリケーション1で他のドライバが指定された場合は、Windowsに備わっているプリントプロセッサ13が呼び出される。

【0036】CAPCALプリントプロセッサ21は、システムスプールキュー16からジョブ記述ファイル18と出力制御ファイル17の内容を取り込み、CAPCALスプールキュー22にCAPCALジョブ記述ファイル23として書き込む。ここで、CAPCALクライアント定義ファイル20には、出力制御情報のデフォルト値が定義されており、出力制御ファイル17に出力制御情報が格納されていない場合は、CAPCALクライアント定義ファイル20から出力制御情報を取り込む。

【0037】また、CAPCALプリントプロセッサ21は、システムスプールキュー16のEMFファイル19を、そのままCAPCALスプールキュー22のEMFファイル24に書き込む。CAPCALジョブ記述ファイル23及びEMFファイル24が、CAPCALスプールキュー22へ書き込まれると、CAPCALプリントプロセッサ21は、このことをCAPCALジョブ転送スレッド27に通知する。

【0038】CAPCALジョブ転送スレッド27は、予め定義された転送手段に基づいて、CAPCALジョブ記述ファイル23及びEMFファイル24をサーバ301に転送する。なお、転送手段及び転送先のサーバ名は、CAPCAL転送先定義ファイル25に格納されており、CAPCALジョブ転送スレッド27は、CAPCAL転送先定義ファイル25の内容に従って、転送手段及び転送先のサーバを決定する。なお、転送手段として、ファイル転送プロトコル(FTP)を用いた通信ネットワーク28やメールの添付ファイルを用いた通信ネットワーク29が使用可能である。また、これら以外の転送手段を用いても良い。

【0039】このように、仮想ドライバとしてのCAPCALドライバ14をクライアント制御部4に設ける。そして、ジョブ記述ファイル18及びEMFファイル19を、システムスプールキュー16にGDI2によりスプールさせるとともに、出力制御ファイル17をCAPCALドライバ14が独自にシステムスプールキュー16にスプールする。このことにより、ジョブ記述ファイル18及びEMFファイル19の形式を、Window

s 標準の形式と一致させることができるとなり、W i n d o w s のアプリケーションインターフェースを一切変更せずに、汎用パッケージソフトを含むW i n d o w s 上で動作する全てのアプリケーションについて、ドキュメント配布処理、分散印刷処理、配布先自動検出処理、プリンタエラー及び印刷完了の通知処理等を実現することができる。この結果、ドキュメント配布処理、分散印刷処理、配布先自動検出処理、プリンタエラー及び印刷完了の通知処理等を実現するために、印刷用のアプリケーションを新たに構築する必要がなくなる。

【0040】図3は、図1のサーバ制御部の構成を説明するブロック図である。図3において、サーバ制御部6には、メール用のCAPCAL受信スレッド30、f t p用のCAPCAL受信スレッド31、CAPCALサーバメインプロセス32、CAPCALジョブ記述ファイル34、39、EMFファイル35、40、CAPCALサーバ側定義ファイル36、配布分散処理スレッド37、プリンタ別出力スレッド41、42及びCAPCALジョブ転送スレッド43が設けられている。

【0041】CAPCALサーバメインプロセス32によりCAPCAL受信スレッド30、31が起動されると、CAPCAL受信スレッド30、31は、CAPCALジョブ記述ファイル23及びEMFファイル24をCAPCALジョブ転送スレッド27から受信する。そして、受信したCAPCALジョブ記述ファイル23及びEMFファイル24の内容を、CAPCALスプールドキュメントキュー33のCAPCALジョブ記述ファイル34及びEMFファイル35に書き込む。ここで、CAPCAL受信スレッド30、31は、ファイル名(ジョブID)が重複しないように管理を行う。CAPCALスプールドキュメントキュー33への書き込みが完了したら、CAPCALサーバメインプロセス32は配布分散処理スレッド37を起動し、CAPCALスプールドキュメントキュー33への書き込み完了を、配布分散処理スレッド37に通知する。

【0042】配布分散処理スレッド37は、CAPCALジョブ記述ファイル34から出力制御情報を取り込む。また、CAPCALサーバ側定義ファイル36の内容を予め読み込んでおく。そして、配布分散処理スレッド37は、出力制御情報及びサーバ側定義ファイルの内容から、出力プリンタ及び出力する開始頁、終了頁を決定し、プリンタ9ごとに設けられたCAPCALスプールプリントキュー38に印刷要求を投入する。CAPCALサーバ側定義ファイル36には、配布先名とプリンタ名(またはサーバ名)との対応関係が格納されており、配布先が論理名で特定されている場合においても、CAPCALサーバ側定義ファイル36を参照することにより、出力プリンタを決定することができる。CAPCALスプールプリントキュー38へのスプールが完了したら、印刷要求があったプリンタ9を専門に受け持つ

プリンタ別出力スレッド41に通知する。

【0043】プリンタ別出力スレッド41は、CAPCALスプールプリントキュー38からEMFファイル40を取り出し、GDI7を呼び出すことにより、W i n d o w s 印刷制御5に描画データを送る。なお、出力制御情報及びCAPCALサーバ側定義ファイル36の内容により、出力プリンタとして、サーバ301が管理しているプリンタ9以外のプリンタが指定されている場合、配布分散処理スレッド37は、CAPCALジョブ転送スレッド43から通信ネットワーク44を介し、指定されているプリンタを管理しているサーバに印刷要求を送る。このように、CAPCALサーバ側定義ファイル36に、配布先名とプリンタ名(またはサーバ名)との対応関係を格納しておくことにより、配布先が論理名で特定されている場合においても、出力プリンタを決定することが可能となり、ユーザは、出力プリンタ名を意識することなく、配布印刷を行うことが可能となる。

【0044】また、プリンタ別出力スレッド41、42を各プリンタごとに設け、プリンタ別出力スレッド41、42がエミュレーションを行うことにより、プリンタの仕様が異なる場合においても、適切な出力結果を得ることが可能となり、プリンタ機種やプリンタメーカーを意識することなく、ネットワーク上の様々なプリンタを利用して印刷を行うことが可能となる。

【0045】図4は、図1のプリンタ出力制御部の構成を説明するブロック図である。図4において、プリンタ出力制御部8には、セントロポートモニタ48及びプリンタ出力管理部50が設けられ、サーバ制御部6には、CAPCALサーバメインプロセス32、EMFファイル40及びプリンタ別出力スレッド41が設けられ、W i n d o w s 印刷制御部5には、Rawスプールファイル46、プリントプロセッサ47及びスプーラープロセスマイン49が設けられ、クライアント300には、CAPCALクライアントメインプロセス26が設けられている。

【0046】プリンタ別出力スレッド41は、CAPCALスプールプリントキュー38からEMFファイル40を取り出し、GDI7を呼び出すことにより、W i n d o w s 印刷制御部5に描画データを送る。GDI7は、印刷指定されたプリンタ9用のプリンタドライバ45と交信し、印刷指定されたプリンタ9専用のコマンド(ESCシーケンス)にEMFファイル40を変換する。変換されたESCシーケンスは、W i n d o w s のシステムスプールキューに格納される。これをRawスプールファイル46と言う。Rawスプールファイル46は、純粋なプリンタコマンド(ESCシーケンス)だけで構成される。Rawスプールファイル46は、W i n d o w s のスプーラープロセスマイン49によりプリントプロセッサ47で読み込まれ、1頁分の描画データ単位で、セントロポートモニタ48に送られる。

【0047】プリンタ9とサーバ302とのインターフェースがLANの場合、LANポートモニタに送られ、その他のインターフェースの場合、そのインターフェース用のポートモニタが送られる。なお、ここまで処理は、Windows印刷制御部5によるデータ処理と同一である。

【0048】セントロポートモニタ48は、スプーラプロセスマイン49経由で受け取ったプリンタ9のコマンドデータを、セントロポートに出力する。ここで、プリンタ9にエラーが発生し、コマンドデータを出力できなかった場合、セントロポートモニタ48は、プリンタ出力管理部50に、プリンタ名称、エラーコード及び元のドキュメントを識別するジョブIDを通知する。また、セントロポートモニタ51も、そのセントロポートモニタ51に接続されているプリンタにエラーが発生した場合、プリンタ出力管理部50に、プリンタ名称、エラーコード及び元のドキュメントを識別するジョブIDを通知する。ここで、ジョブIDは、Windows印刷制御の仕組みに存在するプリントジョブを特定するIDであり、ジョブIDをプリンタ出力管理部50に通知することにより、プリンタ出力管理部50は、元のドキュメント名称、印刷要求したクライアントホスト名及びユーザ名を取得することができる。

【0049】プリンタ出力管理部50は、プリンタ名称、エラーコード及びジョブIDをセントロポートモニタ48から受け取ると、CAPCALサーバメインプロセス32に、これらの情報を通知する。

【0050】CAPCALサーバメインプロセス32は、取得したジョブIDから印刷要求を発行したクライアントホスト名を検出し、そのクライアントホスト上で稼働するCAPCALクライアントメインプロセス26にドキュメント名称、ユーザ名及びエラーコードを通知する。また、画面呼び出し52を行うことにより、これらの情報を表示する。

【0051】CAPCALクライアントメインプロセス26は、CAPCALサーバメインプロセス32からドキュメント名称、ユーザ名及びエラーコードを受信すると、画面呼び出し53を行うことにより、ユーザ名、印刷中のドキュメント名称、エラーメッセージを表示する。

【0052】また、セントロポートモニタ48は、1つのジョブの出力が完全に終了したら、プリンタ出力管理部50にプリンタ名称及びジョブIDを通知する。プリンタ出力管理部50は、セントロポートモニタ48からプリンタ名称及びジョブIDを受け取ると、CAPCALサーバメインプロセス32に、これらの情報を通知する。

【0053】CAPCALサーバメインプロセス32は、取得したジョブIDから印刷要求を発行したクライアントホスト名を検出し、そのクライアントホスト上で

稼働するCAPCALクライアントメインプロセス26に、ドキュメント名称及びユーザ名を通知する。また、画面呼び出し52を行うことにより、これらの情報を表示する。

【0054】CAPCALクライアントメインプロセス26は、CAPCALサーバメインプロセス32からドキュメント名称及びユーザ名を受信すると、画面呼び出し53を行うことにより、ドキュメントの印刷が完了したことを示すメッセージを表示する。

【0055】このように、セントロポートモニタ48は、元のドキュメントを識別するジョブIDをプリンタ出力管理部50に通知することにより、プリンタ出力管理部50は、印刷要求したクライアントホスト名を認識することが可能となり、印刷要求したクライアント300に対し、印刷エラーや印刷完了を通知することが可能となることから、ドキュメントの印刷状況をクライアント300側で認識することが可能となる。

【0056】以下、図1の印刷システムについて、より詳細に説明する。図5は、仮想ドライバ14の動作を説明するブロック図であり、CAPCALスプールファイル57が作成されるまでの処理の流れを示す。

【0057】図5において、ユーザが、アプリケーション1に仮想ドライバ14に対しての印刷を指定した場合、アプリケーション1は、GDI2に対して印刷命令を実行する(CAPCALの場合は自動的にスプール印刷となる)。GDI2は、印刷命令が実行されると、デバイス管理情報(Windowsシステム概値。プリンタドライバ独自の情報を付加可能。以下、DEVMODE情報と記述する)を仮想ドライバ14に要求する。仮想ドライバ14は、メーカやプリンタ機種に依存しない共通な情報(用紙サイズ、用紙種等)と、出力制御ファイル17の名称をDEVMODE情報に設定し、GDI2に返す。GDI2は、仮想ドライバ14から必要な情報を受け取ると、DEVMODE情報と描画情報(GEINFO情報:プリンタドライバ描画能力。プリンタドライバがGDI2に返す情報。Windowsシステム概値)によって、EMFファイル19とジョブ記述ファイル18を作成する。このように、CAPCAL制御印刷では、GDI2が、DEVMODE情報を仮想ドライバ14から受け取ることにより、Windowsシステムが、ジョブ記述ファイル(Windowsシステム概値)18を作成する。

【0058】一方、仮想ドライバ14は、仮想ドライバ14のプリンタプロパティ(指定配布、自動配布、分散印刷の情報を指定)や、予め保存してある出力制御情報により、DEVMODE情報に付加されたファイル名称で出力制御ファイル17を作成する。Windowsのスプールシステム54は、指定されたポートが印刷可能状態であることを確認して、デスプール処理を開始する。デスプール処理により、ポートスレッド55を通じ

てCAPCALプリントプロセッサ21に制御が渡る。CAPCALプリントプロセッサ21は、Windowsシステムが生成したジョブ記述ファイル18を読み込むことにより、DEVMODE情報、仮想ドライバ3によりDEVMODE情報に付加された出力制御ファイル17の名称、及びEMFファイル19の名称を取得する。取得した出力制御ファイル17の名称及びEMFファイル19の名称に基づいて、ジョブ記述ファイル18及びEMFファイル19から出力制御情報と印刷データをそれぞれ取得し、CAPCALスプールファイル57(CAPCALジョブ記述ファイル23及びEMFファイル24)を作成する。

【0059】CAPCALプリントプロセッサ7は、CAPCALスプールファイル57を作成すると、CAPCALスプールファイル57の内容をCAPCALジョブ転送スレッド27を介してサーバ301に送ることにより、指定配布印刷、自動配布印刷、分散印刷等の機能を実現する。

【0060】このように、仮想ドライバ14は、設定されたDEVMODE情報をGDI2に送り、GDI2によりジョブ記述ファイル18とEMFファイル19とを生成させるとともに、出力制御情報を出力制御ファイル17に独自に格納し、独自に格納した出力制御情報をCAPCALプリントプロセッサ21に渡すようにしている。

【0061】この結果、アプリケーション1により生成された描画データを、オペレーティングシステム2によりスプールさせることができとなることから、そのオペレーティングシステム2上で動作するアプリケーション1ならば、どのようなアプリケーション1も使用することが可能となるとともに、ジョブ記述ファイル18及びEMFファイル19とは別個に出力制御情報を格納することにより、オペレーティングシステム2に備わっていない出力制御機能を、後から自由に追加することが可能となり、ネットワーク環境でのドキュメント配布処理、分散印刷処理、配布先自動検出処理、またはプリンタエラー及び印刷完了の通知処理等を、汎用アプリケーションで作成した文書に対して行うことが可能となる。

【0062】図6は、ジョブ記述ファイル18の内容例を示す図である。図6において、ジョブ記述ファイル18は、ジョブ記述ファイルヘッダ61、DEVMODE情報62及びジョブ記述ファイルフッタ63を備えている。なお、これらの情報は、Windowsシステム概値である。DEVMODE情報62は、仮想ドライバ14のDEVMODE構造体情報によって作成される。また、DEVMODE情報62には、プリンタドライバプライベート情報64が仮想ドライバ14により付加される。DEVMODE情報62として、用紙サイズ、用紙種、用紙方向などがあり、プリンタドライバプライベート情報64として、出力制御ファイル名などがある。

【0063】図7は、出力制御ファイル17の内容例を示す図である。図7において、出力制御ファイル17は、CAPCAL出力制御ファイルヘッダ71及び出力制御情報75を備え、出力制御情報75は、動作モード情報72、文書付加情報73及び文書属性情報74を備えている。CAPCAL出力制御ファイルヘッダ71は、CAPCAL出力制御ファイル17の情報を含み、バージョン情報などが格納されている。動作モード情報72は、動作モード(通常モード/配布モード/分散モード)が格納されている。また、仮想ドライバ(CAPCALドライバ)14により指定される配布先情報を含み、配布モード(自動配布/自動配布)、配布先の名称、検索ページ、検索モード(区切りモード/修飾モード/属性モード/サイズモード/色モード/矩形モード)、修飾モードの修飾の種類(下線/網掛け)、属性モードの属性種類(標準/斜体/太字/太字斜体)、サイズモードの文字サイズ、色モードの色コード、及び矩形モードの領域などが格納されている。

【0064】文書付加情報73は、仮想ドライバ14により指定される文書の付加情報を含み、フォーム付加フラグ、バナー付加フラグ、フォームファイルの場所、フォームファイル名、及びバナー挨拶文などが格納されている。

【0065】文書属性情報74は、仮想ドライバ14により指定される文書の属性情報を含み、パスワードフラグ、出力モード(即印刷/日時指定印刷/時間指定印刷/遅延印刷/印刷しない)、日時指定印刷の指定日時、時間指定印刷の指定時間、保存モード(日時指定保存/時間指定保存/無条件保存/保存しない)、及び日時指定保存の指定時間などが格納されている。

【0066】図8は、仮想ドライバ14の動作を示すフローチャートであり、仮想ドライバ14によりジョブ記述ファイル18と出力制御ファイル17とを作成する処理を示すものである。この処理では、ユーザがCAPCALプリンタドライバプロパティを操作するフェーズ(ステップS1~S3)と、仮想ドライバ14の印刷フェーズ(ステップS4~S6)とが設けられている。

【0067】図8において、まず、一般のWindowsプリンタドライバと同様に、CAPCALプリンタドライバプロパティを開く(ステップS1)。なお、CAPCALプリンタドライバプロパティは、プリントフォルダやアプリケーションから開くことができる。

【0068】次に、CAPCALプリンタドライバプロパティが開かれると、ドライバ出力制御設定画面の表示に従って出力制御情報75を設定し、DEVMODE情報62及び出力制御情報75をレジストリに保存する(ステップS2)。そして、アプリケーション1より印刷ダイアログを開き、印刷を実行する(ステップS3)。

【0069】次に、仮想ドライバ14は、CAPCAL

プリンタドライバプロパティからDEVMODE情報62及び出力制御情報75を取得する(ステップS4)。なお、CAPCALプリンタドライバプロパティにDEVMODE情報62及び出力制御情報75が保存されていなければ、仮想ドライバ14に設定されている値を使う。

【0070】次に、仮想ドライバ14は、CAPCALプリンタドライバプロパティから取得したDEVMODE情報(出力制御ファイル名を含む)62をGDI2に渡す(ステップS5)。

【0071】次に、仮想ドライバ14は、CAPCALプリンタドライバプロパティから取得した出力制御情報75を、DEVMODE情報62に付加されている出力制御ファイル名で出力制御ファイル17に書き込む(ステップS6)。

【0072】このように、仮想ドライバ14が、プリンタ機種やメーカーに依存しない共通な情報(DEVMODE情報62)をWindowsシステム(GDI2)に返し、ジョブ記述ファイル18をGDI2に作成させることにより、印刷に必要な情報(用紙種、給紙方式、用紙方向等)を与えるようにしている。また、仮想ドライバ14が、出力制御ファイル17を独自に作成し、配布分散処理に必要な出力制御情報75をCAPCALプリンタプロセッサ21に与えている。このため、汎用のアプリケーション1で作成した文書について、メーカやプリンタ機種に依存しない配布印刷処理を実現することが可能となる。すなわち、ユーザは、ネットワーク上のどのプリンタに印刷させる場合も、仮想ドライバ14で印刷を実行すれば良く、仮想ドライバ14が与える情報によって、プリンタ機種やメーカーで特定した機能を意識することなく、ネットワーク上の多種多様なプリンタ9を利用できる。

【0073】さらに、アプリケーション1で作成したドキュメントは、出力装置(プリンタ9)の物理仕様に依存せず、仕様が異なるプリンタに出力しても期待する結果を得ることが可能となることから、可搬性を向上させることができる。この結果、システム上の全てのアプリケーション1の出力ドキュメントを処理対象として、物理的な出力装置の個々の仕様に依存しない論理デバイスドライブ方式を提供することが可能となる。また、出力装置の機能が不足する場合も、適切なエミュレートで期待する出力を得ることができる。

【0074】図9は、DEVMODE情報の設定画面を示す図である。CAPCALプリンタドライバプロパティを開いた際に、このDEVMODE情報の設定画面を表示させることにより、給紙位置、用紙サイズ、印刷用紙、拡大/縮小、用紙方向、紙種、コピー枚数などを入力することができ、図6のDEVMODE情報62を設定することができる。

【0075】図10は、自動配布情報の設定画面を示す

図である。CAPCALプリンタドライバプロパティを開いた際に、この自動配布情報の設定画面を表示させることにより、検索ページ、印刷時指定、検索モード(区切りモード/修飾モード/サイズモード/色モード/属性モード)、検索範囲などを入力することができ、図7の出力制御情報75を設定することができる。自動配布とは、ユーザが配布先を指定しなくとも、配布される文書の中から配布先を自動的に抽出して配布を行うものである。ここで、検索ページは、配布先を抽出する文書のページを示す。区切りモードでは、指定された文字列で挟まれた領域が検索範囲とみなされ、例えば、括弧を指定すると、括弧で挟まれた文字列が配布先となる。修飾モードでは、修飾された文字列が配布先として検出され、サイズモードでは、指定された大きさを有する文字列が配布先とみなされ、色モードでは、指定された色を有する文字列が配布先とみなされ、属性モードでは、指定された属性の文字列が配布先とみなされる。

【0076】また、検索範囲の入力は、この検索範囲を1枚の仮想的な用紙とみなし、用紙を分割して得られた領域を指定することにより、検索範囲を特定するものである。例えば、縦横の中心線で仮想用紙を4分割した場合の右上の領域と指定する。そして、この仮想用紙での領域の指定を実際の用紙に変換する場合、縦横の中心線で実際の用紙を4分割し、4分割して得られた右上の領域を検索範囲とする。なお、用紙の分割の方法は、4分割の他に、8分割や16分割などであってもよい。この検索範囲の入力により、システム上の全てのアプリケーションの出力ドキュメントを処理対象として、アプリケーションから独立した処理でドキュメントの領域を指定可能となるとともに、ドキュメントの形状(用紙サイズ、用紙の置き方)に依存することなくドキュメントの領域を指定できる。

【0077】図11(a)は、本発明の一実施例に係わる領域指定制御部を説明するブロック図である。図11(a)において、仮想ドライバ14のユーザインターフェース81には、領域指定制御部82が設けられている。領域指定制御部82は、ユーザが指定した領域情報を仮想ドライバ14によって取り込み、文書の印刷情報をとして登録する。

【0078】図11(b)は、実際の用紙を表す長方形を表示することにより、ユーザに領域指定させる方法を説明する図である。図11(b)において、用紙の置き方に依存して、ポートレートの用紙83を表す長方形と、ランドスケープの用紙85を表す長方形とが表示される。そして、ユーザは、ポートレートの用紙83及びランドスケープの用紙85のそれぞれに対し、検索範囲を示す領域84、86を指定する。

【0079】図11(c)は、仮想用紙87を表す長方形を表示することにより、ユーザに領域指定させる方法を説明する図である。図11(c)において、用紙を表す

す長方形を、正方形の仮想用紙87としてMMI（マンマシンインターフェース）としている。ユーザは、この仮想用紙87上で矩形領域88を指定する。

【0080】そして、印刷を実行するときに、仮想用紙87上で指定された情報を取り込み、ポートレートの用紙89、及びランドスケープの用紙91のそれぞれの場合について、仮想用紙87上の矩形領域88を実際の用紙に合わせて変換する。仮想用紙87の大きさを、実際の用紙の大きさに対応させて変換する場合、例えば、仮想用紙87の縦横の変倍率に対応させて、仮想用紙87に指定された矩形領域88を変換する。このことにより、仮想用紙87上での矩形領域88の相対的な位置を保存したまま、実際の用紙89、91上での検索範囲を示す領域90、92に変換することができる。

【0081】例えば、ポートレートの用紙89及びランドスケープの用紙91のいずれの場合においても、配布先は、用紙89、91の上方の欄に記入されることが多い。このため、仮想用紙87上の上方の領域を矩形領域88として指定することにより、仮想用紙87がポートレートの用紙89及びランドスケープの用紙91のいずれに変換された場合においても、検索範囲を示す領域90、92が用紙89、91の上方の欄に位置するように変換され、検索範囲の指定を効率的に行うことが可能となる。

【0082】このように、領域指定機能を、アプリケーション及びドキュメントから独立した階層で提供し、領域指定のMMI（マンマシンインターフェース）を、ドキュメントの形状から独立した正方形の仮想用紙87としたことにより、配布先の検索などのため印刷ドキュメントの領域を指定することが必要な場合、アプリケーションに領域指定機能を組み込むことなく、全ての出力ドキュメントに対して領域設定ができる。この領域設定は、全てのアプリケーションから出力される全てのドキュメントに適用できる。また、領域設定は、ドキュメントから独立しており、1つの領域設定で全ての出力ドキュメントを対象とすることができます。さらに、各ドキュメントの形状（用紙サイズ、置き方）に依存せずに設定することができるので、用紙サイズ、置き方の違いによる領域指定作業の軽減が図れるとともに、一度設定すれば、ドキュメントの形状を変更しても再設定を不要とすることができる。

【0083】図12は、指定配布情報の設定画面を示す図である。CAPCALプリンタドライバプロパティを開いた際に、この指定配布情報の設定画面を表示させることにより、配布先モード（配布先名指定／グループ名指定／プリンタ名指定）、配布先名、グループ名、プリンタ名などを入力することができ、図7の出力制御情報75を設定することができる。例えば、配布先名指定を選択した場合、「東京総務課」などの論理名で印刷物の配布先を指定することが可能となり、プリンタ名を意識

することなく配布先を指定することが可能となる。また、グループ名指定を選択した場合、1回の指定で複数の配布先を選択することができる。さらに、選択された配布先を、画面上で自由に追加したり、削除したりすることができる。配布先の変更も容易に行うことができる。

【0084】図13は、本発明の一実施例に係わる分散印刷情報の設定画面を示す図である。CAPCALプリンタドライバプロパティを開いた際に、この分散印刷情報の設定画面を表示させることにより、分散方法（台数均等／部数別／種別／マニュアル指定）、カラーとモノクロの区別、キューリングの数、プリンタのエラーなどを入力することができ、図7出力制御情報75を設定することができる。分散方法を指定することにより、印刷部数を複数のプリンタに均等に割り振ったり、プリンタの印刷能力や稼働状況に応じて割り振ったりすることができる。また、カラーとモノクロの区別を指定することにより、カラープリンタには、カラーのページだけを出力させたり、モノクロプリンタには、モノクロのページだけを出力させたりすることができる。キューリングの数を指定することにより、出力待ちが貯まっているプリンタを避けて、空いているプリンタを優先させて出力させることができる。プリンタのエラーを指定することにより、配布分散スレッド37がプリンタを決定する時に、エラーが発生しているプリンタを分散対象のプリンタとして扱わないようにすることができる。

【0085】図14は、CAPCALプリントプロセッサ21の構成を示すブロック図である。図14において、印刷データ読み込み制御部101を介して読み込まれた印刷データは、印刷データ格納部102に格納される。印刷データ格納部102に格納された印刷データは、印刷データ構成部103で各ページごとにデータ区分され、CAPCALスプールファイル生成部104に送られる。CAPCALスプールファイル生成部104は、印刷データの各頁ごとの出力による頁管理情報を頁管理情報格納部105に格納し、CAPCALスプールファイル57のEMFファイル24に各頁の印刷データを出力する。全ての印刷データがCAPCALスプールファイル57のEMFファイル24に出力されると、CAPCALジョブ記述ファイル23が作成される。

【0086】CAPCALジョブ記述ファイル23を作成する場合、DEVMODE情報とWindowsジョブ情報とが、印刷ジョブ情報読み込み制御部106から読み込まれ、CAPCAL出力制御情報が、CAPCAL出力制御ファイル読み込み制御部107から読み込まれ、CAPCALユーザ情報が、CAPCALユーザ情報読み込み制御部108から読み込まれる。そして、DEVMODE情報、Windowsジョブ情報、CAPCAL出力制御情報及びCAPCALユーザ情報が、C

APCALジョブ情報格納部109に格納される。印刷ジョブ情報構成部110は、CAPCALジョブ情報格納部109に格納された情報に基づいて、印刷ジョブ情報を構成し、CAPCALスプールファイル生成部105を介してCAPCALスプールファイル57のCAPCALジョブ記述ファイル23に出力する。この際、CAPCALスプールファイル生成部105は、頁管理情報格納部105に格納されている頁管理情報を、印刷ジョブ情報構成部110で構成された印刷ジョブ情報に付加し、CAPCALジョブ記述ファイル23に出力する。

【0087】図15は、CAPCALジョブ記述ファイル23の内容例を示す図である。図15において、CAPCALジョブ記述ファイル23は、CAPCALジョブ記述ファイルヘッダ111、Windowsジョブ情報112、DEVMODE情報113、CAPCAL出力制御情報114、CAPCALユーザ情報115及び印刷データ頁管理情報116を備えている。

【0088】CAPCALジョブ記述ファイルヘッダ111は、CAPCALジョブ記述ファイル23のヘッダ情報として、バージョン番号などの情報を含んでいる。Windowsジョブ情報112は、Windowsのジョブ情報として得られる情報であり、印刷を依頼したユーザ名、印刷ジョブの総頁数、印刷ジョブが発生したホスト名、印刷ジョブタイトル名、印刷ジョブ発生時刻などの情報を含んでいる。

【0089】DEVMODE情報113は、アプリケーション1及び仮想ドライバ14により指定される印刷ジョブのDEVMODE情報113である。CAPCAL出力制御情報114は、図7の出力制御情報75と同一の内容を有している。

【0090】CAPCALユーザ情報115は、CAPCAL用に定義されたユーザの情報であり、氏名、会社名、部署名、電話番号、FAX番号、パスワードなどの情報を含んでいる。

【0091】印刷データ頁管理情報116は、印刷データのファイル上の各頁の先頭位置を示すインデックス情報である。この印刷データ頁管理情報116を参照することにより、配布印刷や分散印刷を容易に行うことができる。

【0092】図16は、CAPCALプリントプロセッサ21の動作を示すフローチャートである。図16において、まず、CAPCALプリントプロセッサ21は、EMFファイル24の作成を行う（ステップS11～S14）。具体的には、印刷データ読み込み制御部101を介して読み込まれた印刷データを、印刷データ格納部102に格納する（ステップS11）。次に、印刷データ構成部103は、印刷データ格納部102から、1頁分の印刷データを描画単位ごとに取得し、CAPCALスプールファイル生成部104を介して、CAPCAL

スプールファイル57に各頁の印刷データを出力する（ステップS12）。この際、CAPCALスプールファイル生成部104の頁管理情報格納部105には、印刷データ頁管理情報116が格納される。次に、未処理の印刷データがあるかどうかを判断し（ステップS13）、未処理の印刷データがある場合、ステップS12に戻って処理を繰り返す。次に、未読み込みのスプールデータがあるかどうかを判断し（ステップS14）、未読み込みのスプールデータがある場合、ステップS11に戻って処理を繰り返す。

【0093】CAPCALプリントプロセッサ21が、CAPCALスプールファイル57に印刷データを出した後、CAPCALジョブ記述ファイル23の作成を行う（ステップS15～S17）。具体的には、印刷ジョブ情報読み込み制御部106によりWindowsジョブ情報112とDEVMODE情報113とが読み込まれ、CAPCAL出力制御ファイル読み込み制御部107によりCAPCAL出力制御情報114が読み込まれ、CAPCALユーザ情報読み込み制御部108によりCAPCALユーザ情報115が読み込まれる。そして、Windowsジョブ情報112、DEVMODE情報113、CAPCAL出力制御情報114及びCAPCALユーザ情報115が、CAPCALジョブ記述ファイル23生成用の情報として、CAPCALジョブ情報格納部109に格納される（ステップS15）。

【0094】次に、印刷データの各頁ごとの書き込みにより頁管理情報格納部105に格納された印刷データ頁管理情報116を、CAPCALスプールファイル57に書き込む（ステップS16）。

【0095】次に、CAPCALジョブ情報構成部110は、CAPCALジョブ情報格納部109に格納されているWindowsジョブ情報112、DEVMODE情報113、CAPCAL出力制御情報114及びCAPCALユーザ情報115を、CAPCALスプールファイル生成部104を介してCAPCALスプールファイル57に書き込む。

【0096】このように、CAPCALプリントプロセッサ21は、Windowsシステムの印刷制御に要求された印刷データ（スプールファイル）をプリンタ9に印刷出力させずに、CAPCAL制御の印刷データとして独自にスプールすることにより、Windowsシステムの印刷制御とは全く異なるサービスの提供が可能となる。この結果、ドキュメントの頁単位、あるいはドキュメント単位に、様々なサービス（出力先の操作／出力順の操作など）をユーザーに提供することが可能となる。

【0097】ここで、印刷データの作成については、Windowsシステムの印刷制御を通して行っているため、汎用アプリケーションからの印刷についても、独自の印刷制御のサービスを提供することが可能となることから、システム上の全てのアプリケーションの出力ドキ

ュメントを処理対象として、例えば、出力先の追加・変更、頁単位での操作、出力順の操作など様々なサービスを提供することが可能となる。

【0098】次に、本発明の実施例に係わる配布印刷処理について説明する。配布印刷処理は、図3の配布分散処理スレッド37が行う。配布分散処理スレッド37では、CAPCALジョブ記述ファイル23中の出力制御情報75で指示された内容に従って、配布印刷、分散印刷、通常印刷のそれぞれの動作が行われる。配布印刷には、自動配布及び指定配布の2種類がある。自動配布は、配布先を印刷ジョブの内容により自動的に決定するモードである。指定配布は、印刷ジョブの配布先をユーザが予め指定するモードである。これら2つのモードのうち、どちらの動作を行うかは、CAPCALジョブ記述ファイル23中で定義することができる。

【0099】図17は、本発明の一実施例に係わる自動配布処理を説明するブロック図であり、自動配布印刷における配布先一覧を得るまでの概要を示す。図17において、サーバ301が、クライアント300から送られてきたCAPCALジョブ記述ファイル23及びEMFファイル24の内容を受信すると、これらの内容をCAPCALスプールドキュメントキューリスト33のCAPCALジョブ記述ファイル34及びEMFファイル35に格納する。配布分散処理スレッド37は、CAPCALジョブ記述ファイル34の内容を読み出し、配布先候補としての文字列122を印刷ジョブ121から検索するための検索条件124を、CAPCALジョブ記述ファイル34から取得する。この検索条件124には、検索ページ及び検索モード（区切り文字／修飾／属性／サイズ／色／矩形領域内）などがあり、この検索条件124を指定することにより、印刷ジョブ121から配布先候補を抽出する方法を指定することができる。なお、この検索条件124は、クライアント300側において、CAPCALプリンタドライバプロパティを開き、図10の自動配布情報の設定画面を表示させることにより、設定することができる。

【0100】配布分散処理スレッド37は、この検索条件124を元に印刷ジョブ121の検索を行い、検索条件124に適合した文字列122を配布先名として取得するとともに、レジストリ123から配布先定義を取得する。レジストリ123には、配布先定義として、N台分配布先名と出力装置名との対応関係が格納されている。このため、比較部125において、印刷ジョブ121から取得した配布先名と、レジストリ123に定義された配布先名とを比較し、レジストリ123に定義された配布先名と一致する文字列122だけを選択することにより、配布すべき全ての出力装置を自動的に決定することができる。

【0101】図18は、自動配布時における検索条件例を示す図である。図18において、検索モードには、

「区切り文字」、「修飾」、「属性」、「サイズ」、「色」、「矩形領域内」などがある。「区切り文字」は、指定された始端文字列と終端文字列との間の文字列122を配布先候補として抽出するものである。なお、印刷時指定により、区切り文字をそのまま出力させたり、スペースで置き換えて出力させたりすることができる。「修飾」は、下線や網掛けなどにより修飾された文字列122を配布先候補として抽出するものである。なお、印刷時指定により、修飾文字をそのまま出力させたり、修飾を解除して出力させたりすることができる。

「属性」は、標準、斜体、太字、太字斜体などの属性を有する文字列122を配布先候補として抽出するものである。なお、印刷時指定により、属性が与えられた文字をそのまま出力させたり、属性を解除して出力させたりすることができる。「サイズ」は、指定された文字サイズ（ポイント数）を有する文字列122を配布先候補として抽出するものである。「色」は、指定された色コードを有する文字列122を配布先候補として抽出するものである。なお、印刷時指定により、指定された色コードを有する文字列をそのままの色で出力させたり、黒に変換して出力させたりすることができる。「矩形領域内」は、検索領域として指定された矩形内の文字列122を配布先候補として抽出するものである。

【0102】図19は、本発明の一実施例に係わる自動配布処理の動作を示すフローチャートであり、配布分散スレッド37が出力装置名称を決定する概要を示す。図19において、まず、配布分散スレッド37は、CAPCALジョブ記述ファイル34の出力制御情報114から検索条件124を取得する。検索条件124として取得するのは、検索ページ、検索モード、条件詳細である（ステップS21）。検索ページは、検索を行う印刷ジョブ121のページである。検索モードは、検索ページ中のどのデータを検索対象とするかを指定するものである。条件詳細は、図18に示すように、指定された検索条件124により異なる情報が格納されている。

【0103】次に、取得した検索条件124に基づき、印刷ジョブ121の検索を行い、検索条件124に該当する全文字列122を印刷ジョブ121から取得し、配布先の候補とする（ステップS22）。次に、レジストリ123から配布先定義（全配布先）を取得し、配布先定義情報一覧を作成する（ステップS23）。配布先定義には、論理的な配布先名称とそれに対応する物理的な出力装置名称とが対となった情報が、配布対象となる出力先の分だけ登録されている。次に、作成された配布先定義情報一覧の中から、配布先定義を1件だけ取り出す（ステップS24）。

【0104】次に、印刷ジョブ121から取得した配布先候補と、配布先定義から取得した配布先名称とを比較し（ステップS25）、印刷ジョブ121から取得した配布先候補と、配布先定義から取得した配布先名称とが

一致する場合、対応する出力装置名称を出力先一覧テーブルに追加する(ステップS26)。

【0105】次に、配布先定義情報一覧に比較していない配布先定義があれば(ステップS27)、ステップS24に戻り、配布先定義情報一覧から新たな配布先定義を取り出し、比較処理を行う。そして、配布先定義情報一覧の中の配布先定義を全て比較した場合、配布先決定処理を終了する。配布先決定処理が終了すると、作成した出力先一覧を元に、それぞれの出力先に印刷ジョブ121の投入を行い、配布印刷を行う。

【0106】このように、クライアント300側で指定された配布先の検索条件124を、サーバ301側のCAPCALジョブ記述ファイル34の内容として送信し、クライアント300から送られた検索条件124に基づいて、ドキュメントに含まれる予め指定された文字列を自動的に判別することにより、システム上の全てのアプリケーション1の出力ドキュメントを対象として、印刷ジョブ121から配布先を自動的に検出することが可能となる。この結果、印刷ドキュメントごとに配布先を指定する手間を省くことが可能となり、システム環境を(配布先に対応するプリンタ名)を意識することなく、1回の印刷指示により検索条件124に該当する全ての配布先へ自動的に印刷を行うことが可能となる。さらに、ユーザの配布方法の選択に幅を持たせることが可能となり、自由度の高い配布処理を実現することが可能となる。

【0107】図20は、本発明の一実施例に係わる指定配布処理を説明するブロック図であり、指定配布印刷における配布先を得るまでの概要を示す。図20において、サーバ301が、クライアント300から送られてきたCAPCALジョブ記述ファイル23及びEMFファイル24の内容を受信すると、これらの内容をCAPCALスプールドキュメントキュー33のCAPCALジョブ記述ファイル34及びEMFファイル35に格納する。配布分散処理スレッド37は、CAPCALジョブ記述ファイル34の内容を読み出し、ここから配布先情報131を取得する。配布先情報131は、指定配布モードとN個の配布先名称とを備え、指定配布モードには、グループ指定、論理名指定(配布先名指定)及び物理名指定(プリンタ名指定)があり、配布先名称には、グループ名、論理名(プリンタ名)及び物理名(配布先名)がある。なお、この配布先情報131は、クライアント300側において、CAPCALプリンタドライバプロパティを開き、図11の指定配布情報の設定画面を表示させることにより、設定することができる。

【0108】ここで、配布分散処理スレッド37は、配布先情報131の指定配布モードが物理名指定の場合、配布先情報131の配布先名称の物理名から、配布すべき出力装置を決定する。

【0109】一方、配布先情報131の指定配布モード

がグループ名若しくは論理名である場合、配布分散処理スレッド37は、レジストリ132から、配布先定義情報及びグループ定義情報を取得する。レジストリ132には、配布先定義として、N台分の配布先名と出力装置名との対応関係が格納されているとともに、グループ定義として、グループ名及び配布先へのリンク情報が格納されている。このため、配布分散処理スレッド37は、レジストリ132に定義されている情報を基に、配布すべき出力装置を決定することができる。

10 【0110】図21は、本発明の一実施例に係わる指定配布処理の動作を示すフローチャートであり、指定配布において配布分散スレッド37が出力装置名称を決定するまでの概要を示す。

【0111】図21において、配布分散処理スレッド37は、CAPCALジョブ記述ファイル34の出力制御情報114から、配布先情報131として、指定配布モードと配布先名称とを取得する(ステップS31)。

20 【0112】次に、配布分散処理スレッド37は、レジストリ132から、配布先定義情報とグループ定義情報を取得する(ステップS32)。配布先定義には、N台分の配布先名と出力装置名との対応関係が登録されている。グループ定義には、論理的なグループ名とそれに対応する配布先定義へのリンクが登録されている。

【0113】次に、配布先情報131から取得した指定配布モードが「物理名指定」である場合(ステップS33)、配布先情報131で示される配布先名称はそのまま出力先名称となる。このため、配布先情報131の配布先名称をそのまま出力先として、出力先一覧を作成する(ステップS34)。一方、配布先情報131から取得した指定配布モードが「論理名指定」である場合(ステップS35)、配布先情報131から取取得した配布先名称は論理名称である。このため、レジストリ132から取得した配布先定義情報を参照することにより出力装置を決定し、出力先一覧を作成する(ステップS36)。

40 【0114】また、配布先情報131から取得した指定配布モードが「グループ指定」である場合、配布先情報131から取得した配布先名称はグループ名である。このため、レジストリ132から取得したグループ定義情報と配布先定義情報を参照することにより出力装置を決定し、出力先一覧を作成する(ステップS37)。出力先一覧の作成が終了すると、作成した出力先一覧を元にそれぞれの出力先に印刷ジョブの投入を行い、配布印刷を行う。

【0115】このように、クライアント300側で指定された配布先情報131を、サーバ301側のCAPCALジョブ記述ファイル34の内容として送信し、クライアント300から送られた配布先情報131に基づいて、配布先を決定することにより、システム上の全てのアプリケーション1の出力ドキュメントを対象として、

配布出力先をドキュメント単位に任意に指定することが可能となる。

【0116】また、レジストリ123、132に、N台分の配布先名と出力装置との対応関係を登録しておくことにより、1つの論理的な配布先に複数の出力装置を定義することが可能となり、配布先を1つ決定するだけで、複数の出力装置にドキュメントを配布することが可能となるとともに、配布先を物理的な出力装置ではなく、論理的な名称で指定可能となる。このため、希望する配布先を印刷実行時に複数選択したり、印刷ドキュメント毎に配布先を指定したり、配布対象であるプリンタの名称を全て指定したりする手間を省くことが可能となり、配布先に対応するプリンタ名称を意識することなく、1回の印刷指示で条件に該当する全ての配布先へ自動的に印刷を行うことが可能となる。

【0117】また、利用者が自由に配布先決定方式を選択することが可能となり、ドキュメントの内容を解析することにより、配布先を印刷ドキュメントデータから自動的に決定したり、配布先対象とする文字列を直接指定することにより、配布先をドキュメントに対して指定したりすることが可能となる。

【0118】さらに、ドキュメント内容による確実な配布先指定が可能となり、配布先の一時的な変更（配布停止など）を容易、かつ安全に行うことが可能となる。この結果、ユーザによる配布方法選択に幅がある自由度の高い配布処理が可能となり、自由度が高く運用状況に柔軟に対応することができるシステム構築及び運用が可能となる。

【0119】次に、本発明の実施例に係わるドキュメント出力監視処理方式について説明する。図22は、本発明の一実施例に係わるドキュメント出力監視処理部の構成を示すブロック図である。図22において、印刷を実行したことにより、スプールファイル142がクライアント300に生成される。そして、クライアント300に生成されたスプールファイル142の内容が、転送モジュール143を介してサーバ301へ転送される。この際、転送モジュール143は、スプールファイル142の内容をサーバ301へ転送したことを、クライアント送信ログ144に出力する。

【0120】サーバ301は、転送モジュール143から送信されたスプールファイル142の内容を受信モジュール148を介して受け取ると、サーバ301上にスプールファイル147を生成する。この際、受信モジュール148は、サーバ301上にスプールファイル147を生成したことを、サーバ受信ログ149に出力する。

【0121】出力制御モジュール150は、スプールファイル147がサーバ301上に生成されると、出力を開始することを出力開始終了ログ151へ書き出し、印刷制御モジュール152にプリンタでの印刷を開始させ

る。プリンタでの印刷が完了すると、印刷制御モジュール152は、プリンタでの印刷が完了したことを印刷ログ153に出力し、出力制御モジュール150へ印刷完了を通知する。その通知を受けた出力制御モジュール150は、出力が完了したことを出力開始終了ログ151へ書き出す。

【0122】通常の印刷では、上述した処理により、クライアント300からサーバ301を経由したプリンタへの印刷は完了する。一方、クライアント300から指定されたプリンタが他サーバ304に対するものであった場合には、サーバ301から他サーバ304に対し、スプールファイル147の内容の転送が行われる。

【0123】印刷するプリンタが他サーバ304の管轄の場合、サーバ301の出力制御モジュール150は、スプールファイル147の内容を転送モジュール154を介して他サーバ304へ転送する。この際、転送モジュール154は、スプールファイル147の内容を他サーバ304へ転送したことを、サーバ送信ログ155に出力する。

【0124】他サーバ304の受信モジュール158は、サーバ301の転送モジュール154から送信されたスプールファイル147の内容を受け取ると、他サーバ304上にスプールファイル157を生成する。この際、受信モジュール158は、サーバ301からスプールファイル147の内容を受け取ったことを、サーバ受信ログ159に出力する。

【0125】他サーバ304の出力制御モジュール160は、スプールファイル157が生成されると、出力を開始することを出力開始終了ログ161へ書き出し、印刷制御モジュール162にプリンタでの印刷を開始させる。プリンタでの印刷が完了すると、印刷制御モジュール162は、プリンタでの印刷が完了したことを印刷ログ163に出力し、出力制御モジュール160へ印刷完了を通知する。その通知を受けた出力制御モジュール160は、出力が完了したことを出力開始終了ログ161へ書き出す。

【0126】クライアント300からサーバ301への転送及び印刷までの過程は、サーバ301から他サーバ304への転送及び印刷までの過程と同じ制御となる。

40 また、例えば、WANに接続されているサーバ間の転送等、支店から本店のプリンタへ向けて印刷を行う場合において、送受信が可能な限られているマシン間でスプールファイルを転送し、そこから指定のプリンタに印刷可能なサーバへの転送が行われることがある。このように、他サーバ304から別のサーバへの転送がさらに発生する場合は、上述した転送処理が繰り返し行われる。

【0127】図22の処理では、クライアント300→サーバ301→他サーバ304におけるスプールファイル142、147、157の処理過程（結果）を示すログが、クライアント送信ログ144、サーバ受信ログ1

49、出力開始終了ログ151、印刷ログ153、サーバ送信ログ155、サーバ受信ログ159、出力開始終了ログ161、印刷ログ163に出力されたことになる。ログ追跡モジュール145が、これらの個々のログから、これらの順番に情報を取得し、1回の印刷実行に対する情報に関連付けることにより、異なるマシン間での印刷の開始から終了までの過程を全て追尾することが可能となる。

【0128】このように、クライアント300での印刷開始、サーバ301、304での印刷実行までの経過、及び印刷実行結果をそれぞれログに格納し、ユーザが認識可能であるドキュメント出力処理をキーとして、それらの情報が格納された複数のログの情報表示や操作などの処理を可能としている。このことにより、出力装置を制御するサーバがネットワーク上に複数存在するシステムにおいて、利用者が投入した複数の出力装置に対する複数のドキュメント出力処理を、出力装置ではなくドキュメント出力処理を主体に監視（状態監視、結果監視）することが可能となる。この結果、ユーザは、自分が投入した印刷処理を、リアルタイムで監視することが可能となるとともに、エラー発生時に、プリンタを変更したり、頁を指定しての再印刷を行わせたりする指示を、簡単、安全、かつ的確に行うことが可能となる。

【0129】図23は、ログ情報の追跡の流れを示すブロック図である。図23において、クライアント300によりクライアント送信ログ144が生成され、サーバ301によりサーバ受信ログ149、出力開始終了ログ151、印刷ログ153及びサーバ送信ログ155が生成され、他サーバ304によりサーバ受信ログ159、出力開始終了ログ161及び印刷ログ163が生成されている。そして、ログ追跡モジュール145により、クライアント送信ログ144、サーバ受信ログ149、出力開始終了ログ151、印刷ログ153、サーバ送信ログ155、サーバ受信ログ159、出力開始終了ログ161及び印刷ログ163が順次に追跡される。

【0130】図24は、ログファイルに収めるログ情報の内容例を示す図である。図22及び図23には、クライアント送信ログ144、サーバ受信ログ149、出力開始終了ログ151、印刷ログ153、サーバ送信ログ155、サーバ受信ログ159、出力開始終了ログ161及び印刷ログ163の8種類のログが存在するが、これらのログ情報の構造は、以下に示すように、全て共通となっている。

【0131】図24において、「ログ情報の種類」170には、ログ情報の種類を示すコードを格納する。例えば、クライアント送信ログ144には、「ログ情報の種類」170として送信ログ（クライアント）を示すコードを格納し、サーバ受信ログ149、159には、「ログ情報の種類」170として受信ログ（サーバ）を示すコードを格納し、出力開始終了ログ151、161に

は、「ログ情報の種類」170として出力開始終了ログ（サーバ）を示すコードを格納し、印刷ログ153、163には、「ログ情報の種類」170として印刷ログ（サーバ）を示すコードを格納し、サーバ送信ログ155には、「ログ情報の種類」170として送信ログ（サーバ）を示すコードを格納する。

【0132】「サーバ上の識別子」171には、サーバ上でユニークに割り当てられる通し番号を格納する。なお、サーバ上の識別子171は、サーバが受信したときに割り当てる番号である。

【0133】「クライアント上の識別子」172には、印刷を実行したクライアント上でユニークに割り当てられる通し番号を格納する。「クライアントのホスト名」173には、印刷を実行したクライアントのホスト名称を格納する。

【0134】「出力先名（プリンタ等）」174には、印刷の出力先の名称を格納する。例えば、「出力先名（プリンタ等）」174にプリンタ名が格納されると、サーバ経由で印刷が行われる。また、「出力先名（プリンタ等）」174に他サーバ管轄のプリンタが格納されると、サーバ間を経由した印刷が行われる。

【0135】「出力情報」175には、印刷実行時に指定した印刷物固有のジョブタイトル名（文書や表のドキュメント名）177、ユーザ名178、用紙サイズ179、出力部数180、出力頁順181、出力開始頁番号182、出力終了頁番号183、出力終了日時184、出力結果コード185などが格納される。この「出力情報」175は、印刷物の内容を確認する際の表示用の情報として使用される。

【0136】「制御情報」176には、通常／配布／分散等の印刷の種類を示す動作モード186や、通常／日時／遅延等の印刷出力のタイミングを示す出力モード187や保存モード188などが格納される。「制御情報」176は、印刷の実行を制御する情報として使用され、どのような制御下で印刷が行われたかを確認する際に有効となる。

【0137】なお、異なるマシン間でのログ情報は、「クライアント上の識別子」172と「クライアントのホスト名」173とのセットで関連付けることができる。このように、異なる処理でも同一構造のログ情報に格納することにより、処理結果の記憶装置への入出力を共通に扱うことが可能となる。

【0138】図25は、本発明の一実施例に係わるドキュメント出力監視方法を示すフローチャートである。図25において、ログ追跡モジュール145は、ログ追跡を開始するクライアント300上のクライアント送信ログ144からログ情報を獲得する（ステップS41）。

【0139】次に、印刷の実行によりスプールファイル142を転送する送信先のサーバ301のサーバ名称を、クライアント送信ログ144のログ情報から獲得す

る（ステップS42）。

【0140】次に、同じクライアント300のホスト名173と識別子172を持つログ情報を、送信先サーバ301上のサーバ受信ログ149から獲得する（ステップS43）。

【0141】次に、サーバ301の出力開始終了ログ151から、出力開始情報を獲得する（ステップS44）。次に、同じクライアント300のホスト名173と識別子172を持つログ情報を以降のログの中から検索する際の検索開始日時として、出力開始終了ログ151から獲得した出力開始情報の出力終了日時184をセットする（ステップS45）。

【0142】次に、出力開始終了ログ151から、出力終了情報を獲得する（ステップS46）。この出力終了情報は、出力が完了している場合にのみ取得することが可能である。

【0143】次に、出力開始終了ログ151のログ情報の中に出力終了情報が有るならば（ステップS47）、検索終了日時として、出力開始終了ログ151から獲得した出力終了情報の出力終了日時をセットする（ステップS48）。

【0144】次に、出力開始終了ログ151から獲得した出力開始情報の出力先名174がプリンタの場合（ステップS49）、同じプリンタ（出力先）で、かつ同じクライアント300のホスト名173と識別子172を持ち、検索開始日時と検索終了日時との間の出力終了日時184を格納しているログ情報を、印刷ログ153から検索する（ステップS50）。

【0145】サーバ301からプリンタへの印刷が完了していれば（ステップS51）、以上の処理によりログの追尾は終了する。出力開始終了ログ151から獲得した出力開始情報の出力先名174がプリンタではなく、サーバ301から他サーバ304へスプールファイル147が転送されている場合（ステップS49）、サーバ301上のサーバ送信ログ155からログ情報を獲得する（ステップS52）。そして、転送した送信先の他サーバ名称をサーバ送信ログ155のログ情報から獲得し（ステップS53）、ステップS43に戻ることにより、他サーバ304は、サーバ301で行われた処理と同様の処理を開始する。

【0146】以上の処理を、印刷が完了もしくは未だ印刷がされていないと判別するまで繰り返し（ステップS51）、サーバを変えて追尾する。クライアントのホスト名173と識別子172とが同じスプールファイル142について、サーバ301上での複数回印刷を指定しても、出力開始終了ログ151の開始と終了の日時を得ることにより、ログ情報の検索範囲を限定することができる。また、検索範囲を限定することにより、検索時間を短縮することができる。

【0147】このように、異なる処理の結果である情報

に関連付けを行い、その関連づけに基づいて送信元を辿ることにより、送信先での処理の結果を送信元に知らせることが可能となり、クライアント300は、プリンタがどのサーバ301、304で管理されているのかを意識することなく、印刷要求を行うことが可能となるとともに、いずれのサーバ301、304において印刷が行われた場合においても、クライアント300は、印刷状況を把握することが可能となる。

【0148】また、情報の1つに対応する処理が終了した日時を取得することにより、それに続く処理の情報をその日時以降を条件として検索することが可能となる。また、特定の処理においては、処理の開始と終了に対する情報を集めることで、それに続く処理の情報が完了しているかどうかを、それに続く処理の情報を得ることなしに確認することができる。

【0149】さらに、処理の開始と終了の日時を得ることにより、それに続く処理の情報から関連する情報を検索する際に、範囲を指定した検索を行うことが可能となる。

#### 【0150】

【発明の効果】以上説明したように、本発明によれば、描画データ生成手段により生成された描画データを、その描画データ生成手段を管理するオペレーティングシステムによりスプールさせるとともに、描画データの出力を制御する出力制御情報を独自に格納することにより、そのオペレーティングシステム上で動作するアプリケーションならば、どのようなアプリケーションも使用することが可能となるとともに、オペレーティングシステムに備わっていなかった出力制御機能を、後から自由に追加することが可能となり、従来のオペレーティングシステムでは容易に実現することができなかつたネットワーク環境でのドキュメント配布処理、分散印刷処理、配布先自動検出処理、またはプリンタエラー及び印刷完了の通知処理等を、汎用アプリケーションで作成した文書に対して行うことが可能となる。

【0151】また、本発明の一態様によれば、出力制御情報生成手段により生成された出力制御情報に基づいて、描画データに対応した独自のプリンタ管理ファイルを生成することにより、オペレーティングシステムと独立した独自の形式でプリンタ管理ファイルの処理や操作を行うことが可能となる。例えば、オペレーティングシステムとしてWindowsを用いた場合、クライアントとサーバとの間の通信ネットワークとして、LANなどに限定されるのに対し、プリンタ管理ファイルを独自の形式に変換することにより、LAN、WAN、インターネットなどの多種多様な通信ネットワークを利用することができる。

【0152】また、本発明の一態様によれば、クライアントから送られた検索条件に基づいて、ドキュメントに含まれる予め指定された文字列を自動的に検出し、検出

された文字列を配布先定義テーブルに格納されている配布先名と比較することにより、印刷ジョブから配布先を自動的に検出することが可能となるとともに、自動的に検出された配布先に基づいて、出力装置を特定することが可能となり、配布先を指定することなしに、1回の印刷指示を行うだけで検索条件に該当する全ての配布先へ自動的に印刷を行うことが可能となる。

【0153】また、本発明の一態様によれば、仮想的な用紙を分割した矩形領域を、配布先候補の検索領域として指定可能とすることにより、検索領域を用紙の上下左右や真ん中などの方向で指定することが可能となり、仮想的な用紙への1回の領域設定により、各印刷ドキュメントにおいて用紙サイズや置き方などが異なる場合においても、配布先の検索領域を適切に設定することが可能となる。

【0154】また、本発明の一態様によれば、N台分の配布先名と出力装置との対応関係を登録したり、1つの論理的な配布先に複数の出力装置を定義したりすることにより、配布先を1つ指定するだけで、その配布先に対応する複数の出力装置を自動的に決定することが可能となり、配布先を物理的な出力装置ではなく、論理的な名称で指定することが可能となるとともに、1回の指定により複数の出力装置にドキュメントを配布することが可能となる。この結果、希望する配布先を印刷実行時に複数選択したり、印刷ドキュメント毎に配布先を指定したり、配布対象であるプリンタの名称を全て指定したりする手間を省くことが可能となり、配布先に対応するプリンタ名称を意識することなく、1回の印刷指示で条件に該当する全ての配布先へ自動的に印刷を行うことが可能となる。

【0155】また、本発明の一態様によれば、スプールファイルの送信結果、受信結果、出力結果及び印刷結果を印刷要求元情報とともに格納しておくことにより、ネットワーク上でのスプールファイルの転送が行われた場合においても、そのスプールファイルの送信結果、受信結果、出力結果及び印刷結果が、どの印刷要求元に起因して得られたのかを容易に判定することが可能となり、自分の印刷要求がどのサーバのプリンタでどのように処理されたのかを認識することが可能となる。この結果、クライアントは、プリンタがどのサーバで管理されているのかを意識することなく、印刷要求を行うことが可能となるとともに、いずれのサーバにおいて印刷が行われた場合においても、印刷状況を正確に把握することが可能となる。

#### 【図面の簡単な説明】

【図1】本発明の一実施例に係わる印刷システムの構成を示すブロック図である。

【図2】図1のクライアント制御部の構成を説明するブロック図である。

【図3】図1のサーバ制御部の構成を説明するブロック

図である。

【図4】図1のプリンタ出力制御部の構成を説明するブロック図である。

【図5】仮想ドライバの動作を説明するブロック図である。

【図6】ジョブ記述ファイルの内容例を示す図である。

【図7】出力制御ファイルの内容例を示す図である。

【図8】仮想ドライバの動作を示すフローチャートである。

10 【図9】DEVMODE情報の設定画面を示す図である。

【図10】自動配布情報の設定画面を示す図である。

【図11】本発明の一実施例に係わる検索領域の指定方法を説明する図である。

【図12】指定配布情報の設定画面を示す図である。

【図13】本発明の一実施例に係わる分散印刷情報の設定画面を示す図である。

【図14】CAPCALプリントプロセッサの構成を示すブロック図である。

20 【図15】CAPCALジョブ記述ファイルの内容例を示す図である。

【図16】CAPCALプリントプロセッサの動作を示すフローチャートである。

【図17】本発明の一実施例に係わる自動配布処理を説明するブロック図である。

【図18】自動配布時における検索条件例を示す図である。

【図19】本発明の一実施例に係わる自動配布処理の動作を示すフローチャートである。

30 【図20】本発明の一実施例に係わる指定配布処理を説明するブロック図である。

【図21】本発明の一実施例に係わる指定配布処理の動作を示すフローチャートである。

【図22】本発明の一実施例に係わるドキュメント出力監視処理部の構成を示すブロック図である。

【図23】ログ情報の追跡の流れを示すブロック図である。

【図24】ログ情報の内容例を示す図である。

40 【図25】本発明の一実施例に係わるドキュメント出力監視方法を示すフローチャートである。

【図26】従来の印刷方法を説明するブロック図である。

【図27】従来の分散印刷方法を説明するブロック図である。

#### 【符号の説明】

300 クライアント

301、304 サーバ

302、303、28、29、44 通信回線

1 アプリケーション

2、7 GDI

3、5 Windows印刷制御部  
 4 クライアント制御部  
 6 サーバ制御部  
 8 プリンタ出力制御部  
 9 プリンタ  
 11 スプーラプロセス  
 12 プロバイダ  
 13、47 プリントプロセッサ  
 14 仮想ドライバ  
 15 CAPCALドライバプロパティ制御スレッド  
 16 システムスプールキュー  
 17 出力制御ファイル  
 18 ジョブ記述ファイル  
 19、24、35、40 EMFファイル  
 20 CAPCALクライアント定義ファイル  
 21 CAPCALプリントプロセッサ  
 22 CAPCALスプールキュー  
 23、34、39 CAPCALジョブ記述ファイル  
 25 CAPCAL転送先定義ファイル  
 26 CAPCALクライアントメインプロセス  
 27 CAPCALジョブ転送スレッド  
 30、31 CAPCAL受信スレッド  
 32 CAPCALサーバメインプロセス  
 33 CAPCALスプールドキュメントキュー  
 36 CAPCALサーバ側定義ファイル  
 37 配布分散処理スレッド  
 38 CAPCALスプールプリンタキュー  
 41、42 プリンタ別出力スレッド  
 43 CAPCALジョブ転送スレッド  
 45 プリンタドライバ  
 46 Rawスプールファイル  
 48、51 セントロポートモニタ  
 49 スプーラプロセスメイン  
 50 プリンタ出力管理部  
 52、53 画面呼び出し  
 54 スプールシステム  
 55 ポートスレッド  
 57、142、147、157 スプールファイル  
 61 ジョブ記述ファイルヘッダ  
 62 DEV MODE情報  
 63 ジョブ記述ファイルフッタ  
 64 プリンタドライバプライベート情報  
 71 CAPCAL出力制御ファイルヘッダ  
 72 動作モード情報  
 73 文書付加情報  
 74 文書属性情報  
 75 出力制御情報  
 81 ユーザインターフェース  
 82 領域指定制御部  
 83、85 89、91 用紙

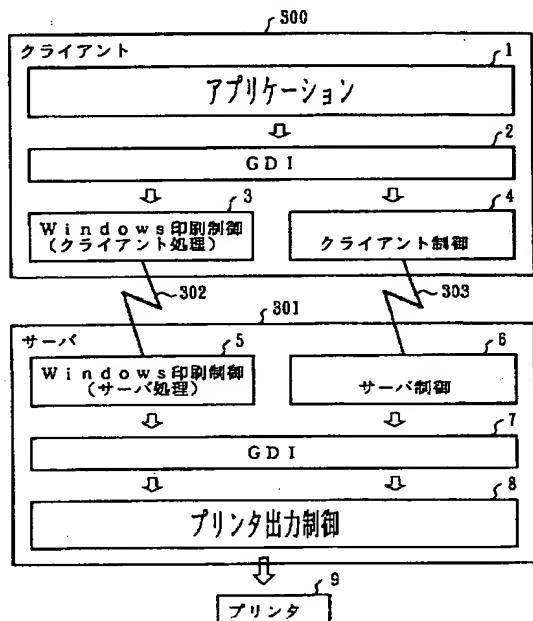
84、86、88、90、92 指定領域  
 87 仮想用紙  
 101 印刷データ読み込み制御部  
 102 印刷データ格納部  
 103 印刷データ構成部  
 104 CAPCALスプールファイル生成部  
 105 頁管理情報格納部  
 106 印刷ジョブ情報読み取り制御部  
 107 CAPCAL出力制御ファイル読み込み制御部  
 108 CAPCALユーザ情報読み込み制御部  
 109 CAPCALジョブ情報格納部  
 110 印刷ジョブ情報構成部  
 111 CAPCALジョブ記述ファイルヘッダ  
 112 Windowsジョブ情報  
 113 DEV MODE情報  
 114 出力制御情報  
 115 CAPCALユーザ情報  
 116 印刷データ頁管理情報  
 121 印刷ジョブ  
 20 122 文字列  
 123、132 レジストリ  
 124 検索条件  
 125 比較部  
 131 配布先情報  
 143、154 転送モジュール  
 144 クライアント送信ログ  
 145 ログ追跡モジュール  
 148、158 受信モジュール  
 149、159 サーバ受信ログ  
 30 150、160 出力制御モジュール  
 151、161 出力開始終了ログ  
 152、162 印刷制御モジュール  
 153、163 印刷ログ  
 155 サーバ送信ログ  
 170 ログ情報の種類  
 171 サーバ上の識別子  
 172 クライアント上の識別子  
 173 クライアントのホスト名  
 174 出力先名(プリンタ等)  
 40 175 出力情報  
 176 制御情報  
 177 ジョブタイトル名  
 178 ユーザ名  
 179 用紙サイズ  
 180 出力部数  
 181 出力頁順  
 182 出力開始頁番号  
 183 出力終了頁番号  
 184 出力終了日時  
 50 185 出力結果コード

186 動作モード

187 出力モード

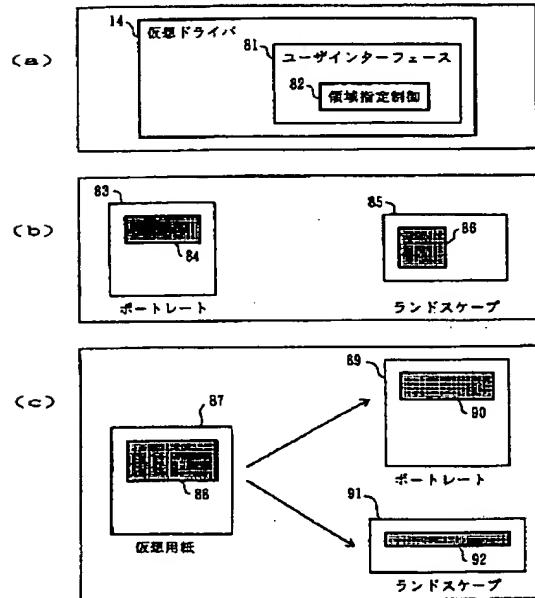
【図1】

本発明の一実施例に係る印刷システムの構成を示すブロック図



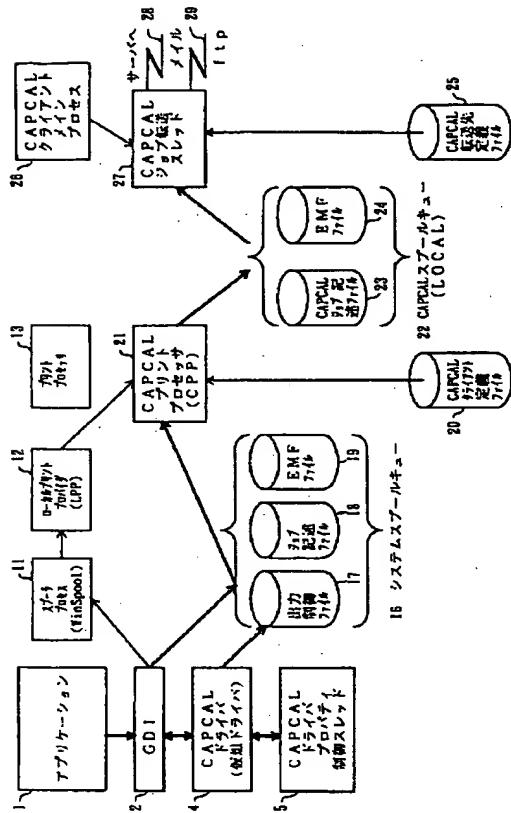
【図1-1】

本発明の一実施例に係る検索領域の指定方法を説明する図



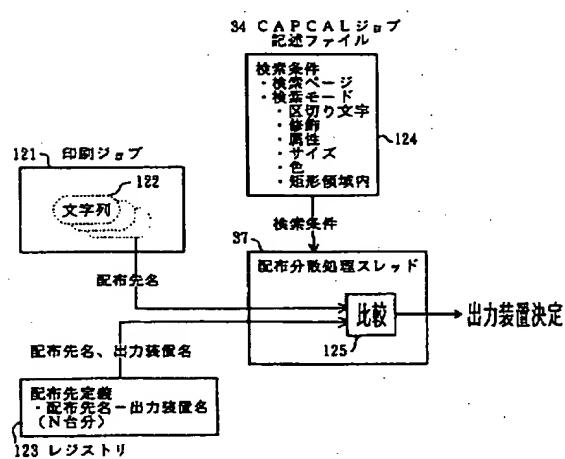
【図2】

図1のクライアント制御部の構成を説明するブロック図



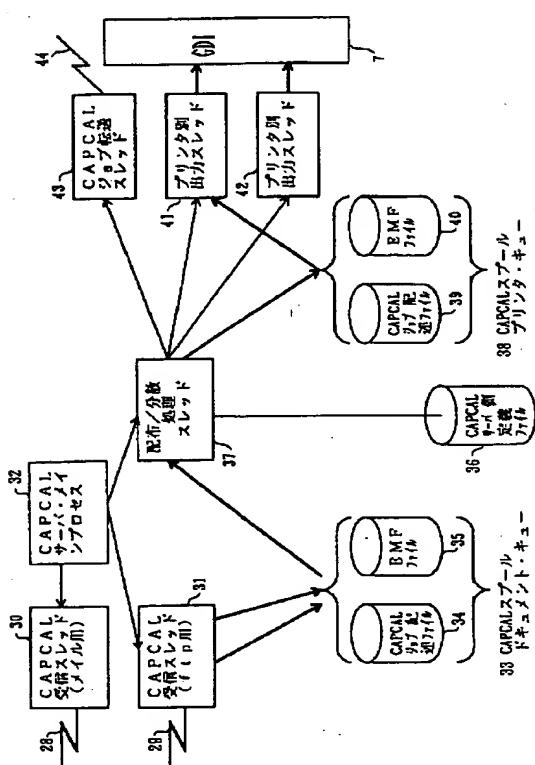
【図1-7】

本発明の一実施例に係る自動配布処理を説明するブロック図



〔四三〕

図1のサーバ制御部の構成を説明するブロック図



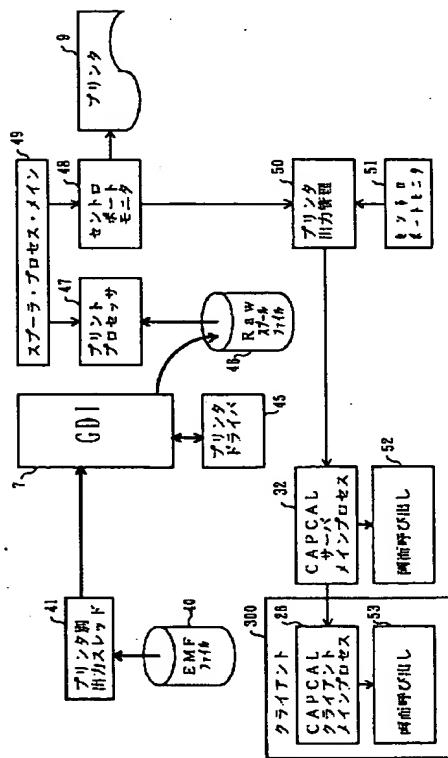
【四】

### 自動配布時における検索条件例を示す図

検索モード	条件詳細
区切り文字	始端文字列 終端文字列 印刷時指定(そのまま出力/スペースで置き換え)
修飾	修飾種(下線/網掛け) 印刷時指定(そのまま出力/修飾解除)
属性	属性種(標準/斜体/太字/太字斜体) 印刷時指定(そのまま出力/属性解除)
サイズ	文字サイズ(点数)
色	色コード 印刷時指定(そのまま出力/黒へ変更)
矩形領域	検索領域を示す矩形の座標

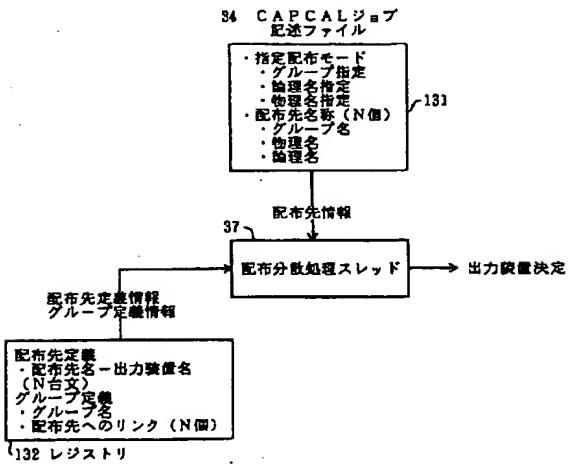
【图 4】

図1のプリンタ出力制御部の構成を説明するブロック図



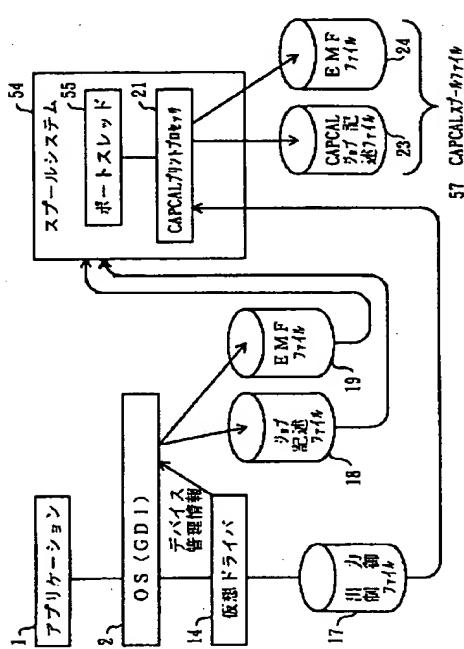
〔四二〇〕

本発明の一実施例に係わる指定配布処理を説明するブロック図



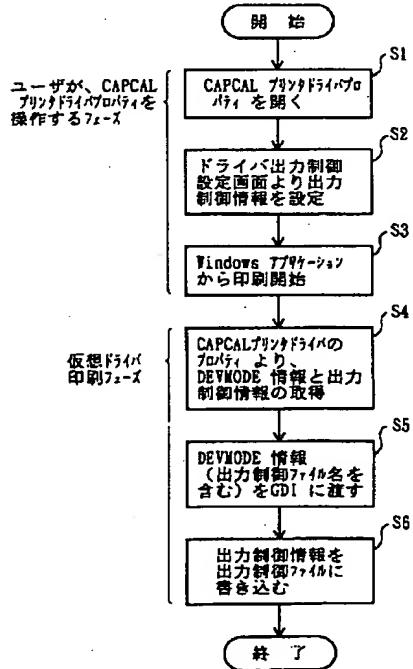
【図5】

仮想ドライバの動作を説明するブロック図



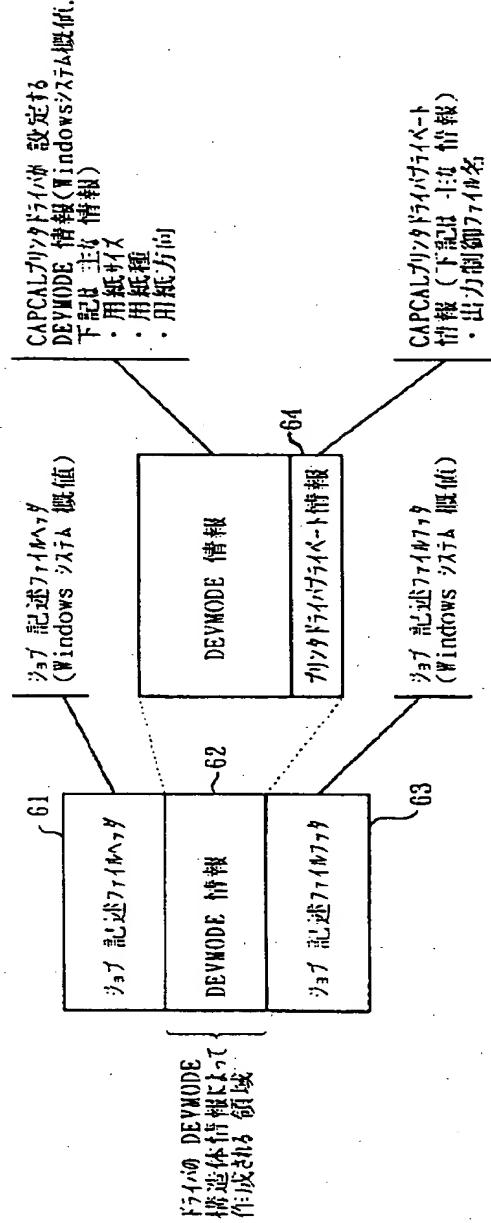
【図8】

仮想ドライバの動作を示すフローチャート



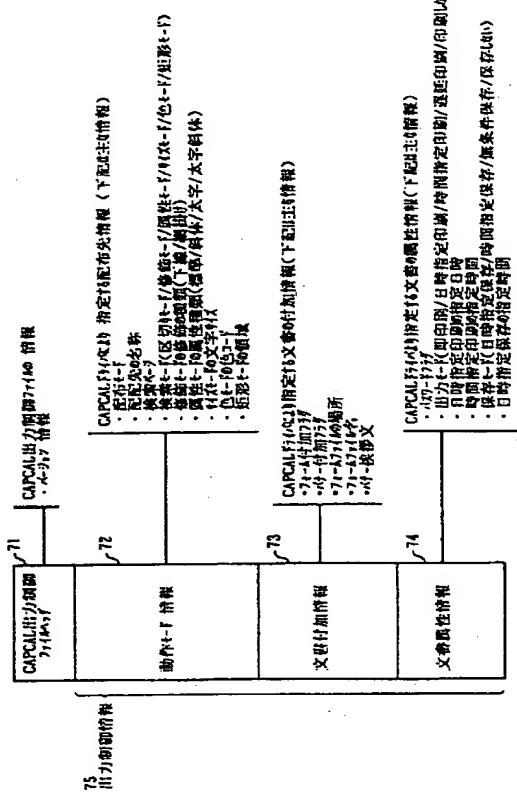
【図6】

ジョブ記述ファイルの内容例を示す図



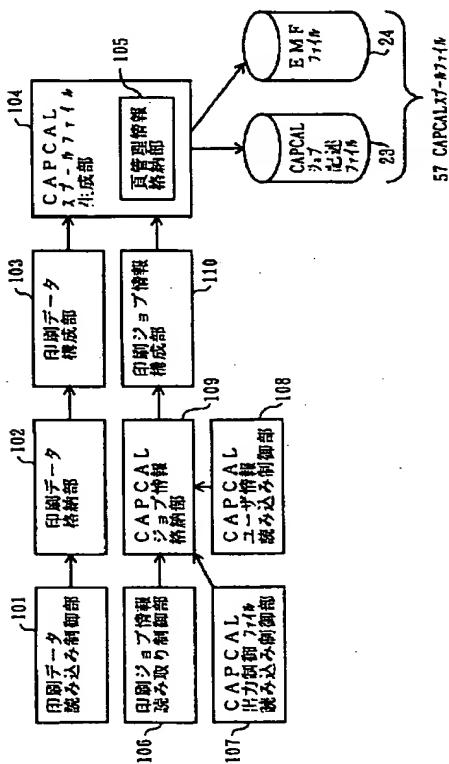
【図7】

出力制御ファイルの内容例を示す図



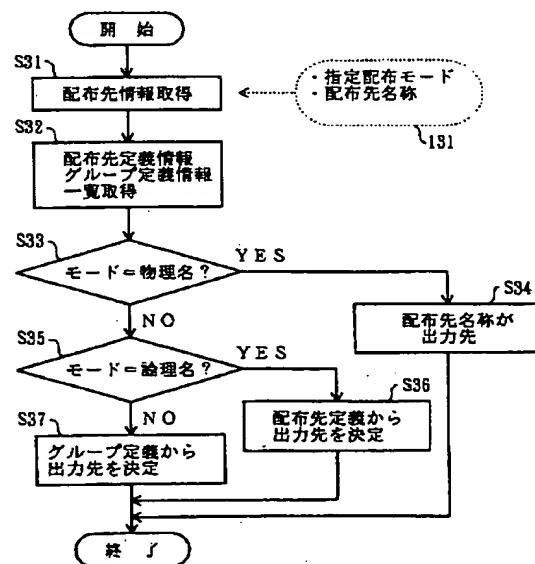
【図14】

CAPCALプリントプロセッサの構成を示すブロック図



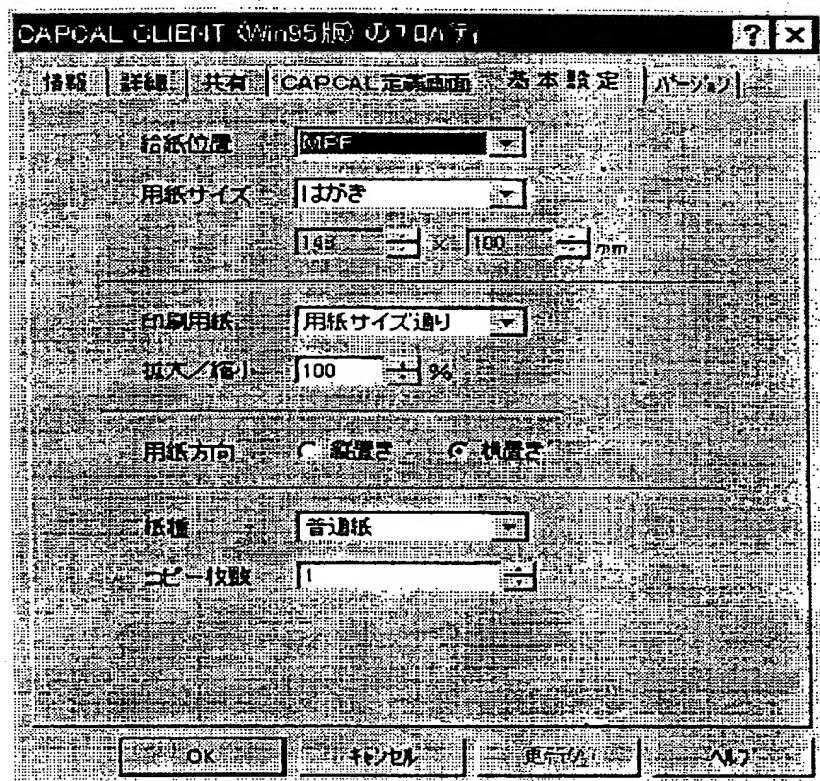
【図21】

本発明の一実施例に係る指定配布処理の動作を示すフローチャート



【図9】

DEVMODE情報の設定画面を示す図



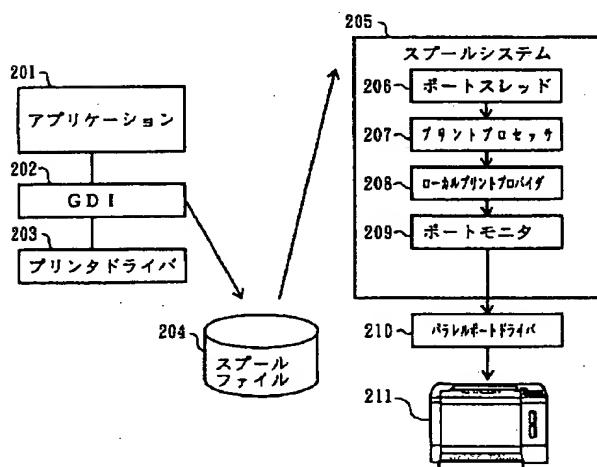
【図24】

ログ情報の内容例を示す図

ログ情報の種類	～170
サーバ上の識別子	～171
クライアント上の識別子	～172
クライアントのホスト名	～173
出力先名(プリンタ等)	～174
出力情報	
ジョブタイトル名	～177
ユーザ名	～178
用紙サイズ	～179
出力部数	～180
出力頁順	～181
出力開始頁番号	～182
出力終了頁番号	～183
出力終了日時	～184
出力結果コード	～185
制御情報	
動作モード	～186
出力モード	～187
保存モード	～188

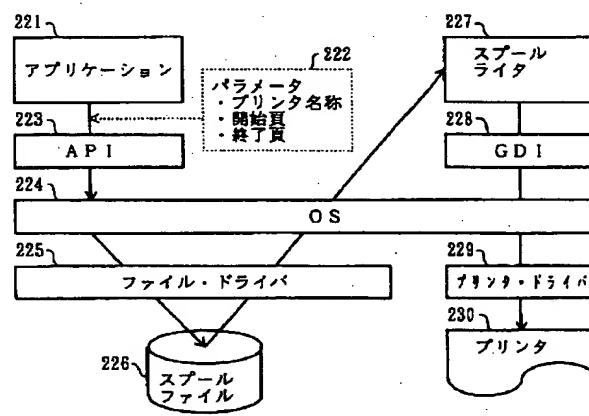
【図26】

従来の印刷方法を説明するブロック図



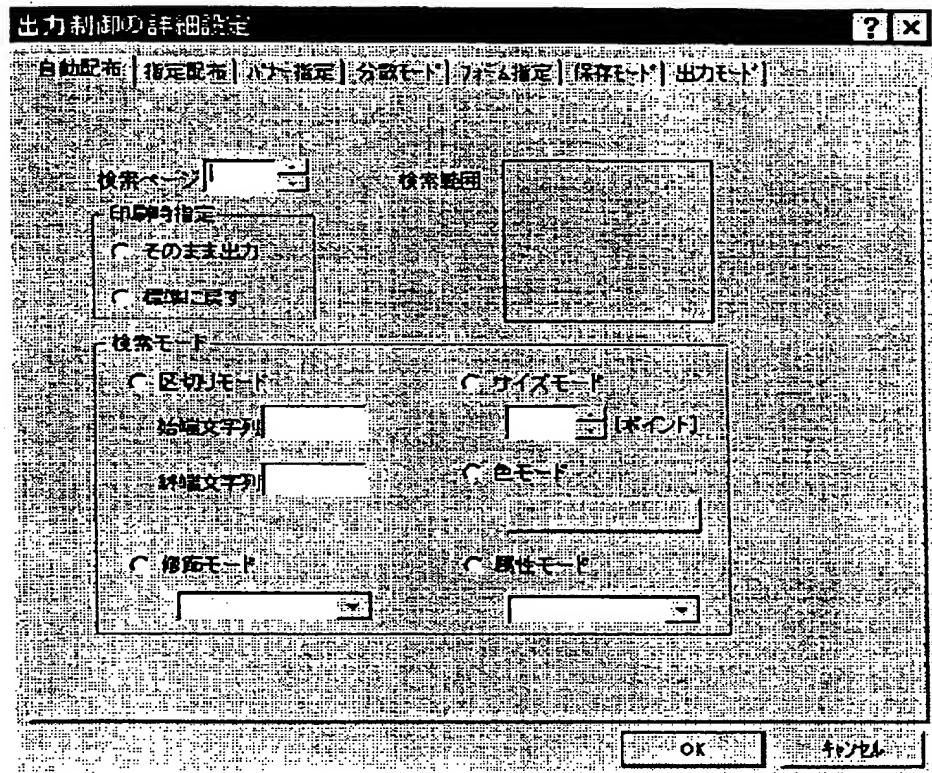
【図27】

従来の分散印刷方法を説明するブロック図



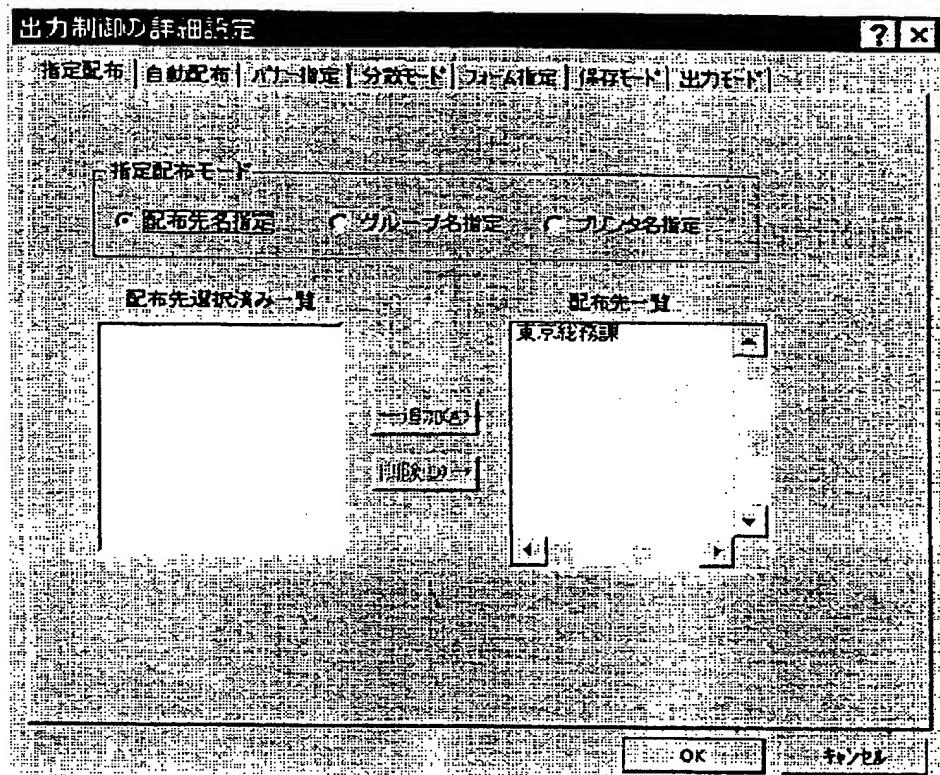
【図10】

## 自動配布情報の設定画面を示す図



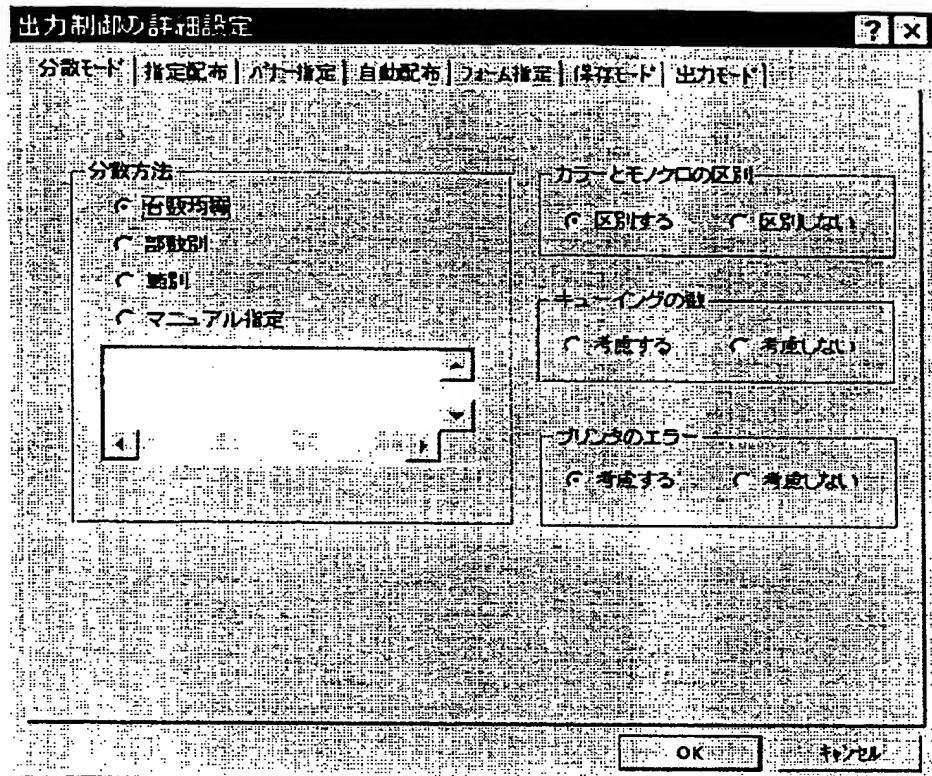
【図12】

## 指定配布情報の設定画面を示す図



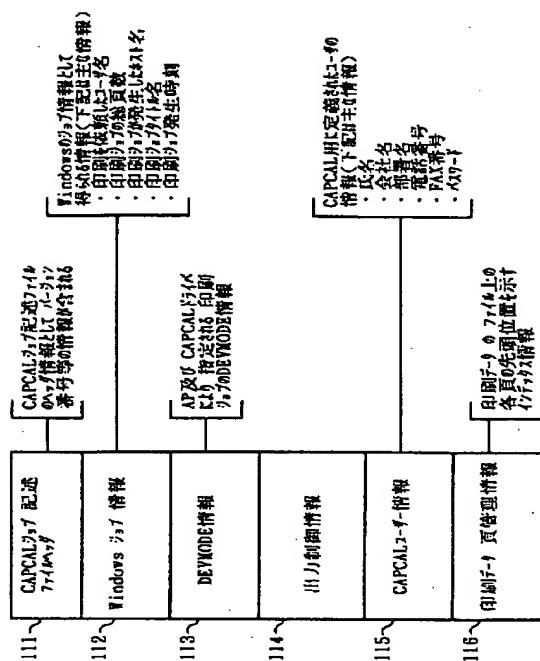
【図13】

本発明の一実施例に係わる分散印刷情報の設定画面を示す図



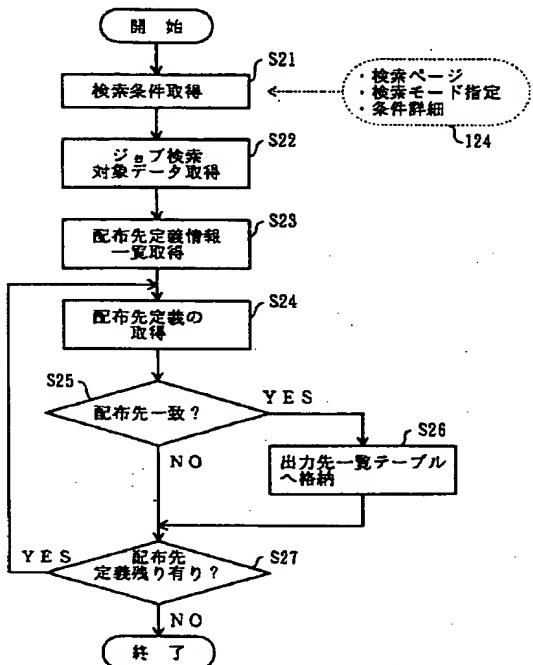
【四】

CAPCALジョブ記述ファイルの内容例を示す図



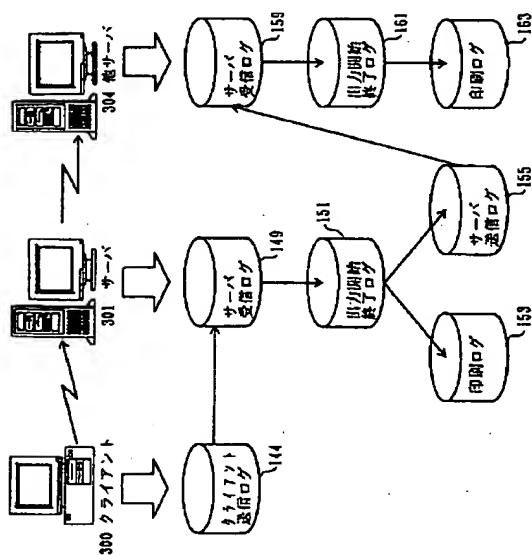
〔 19〕

本発明の一実施例に係る自動配布処理の動作を示すフロー・チャート



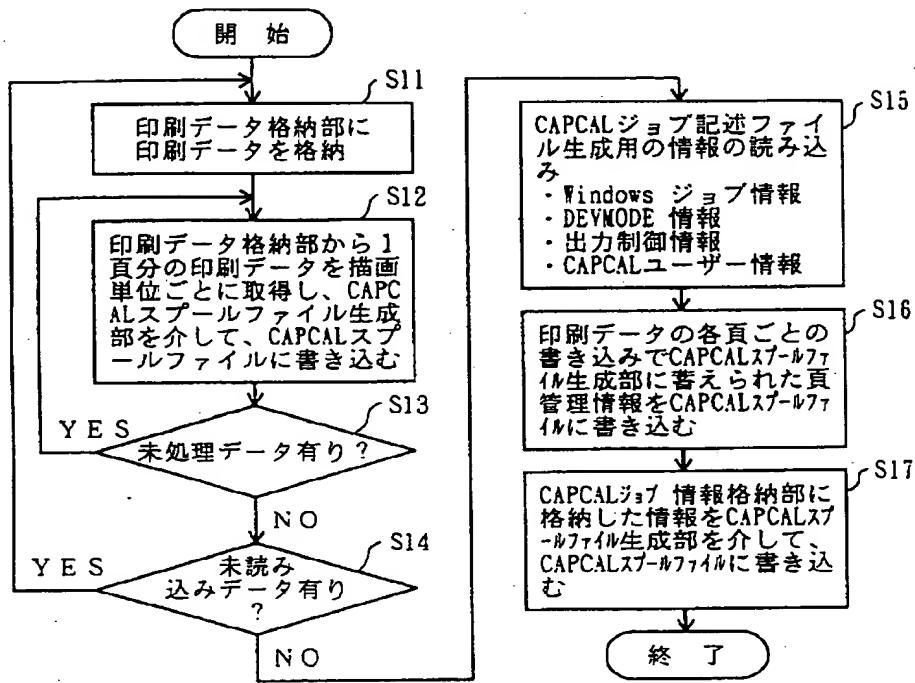
【23】

ログ情報の追跡の流れを示すブロック図



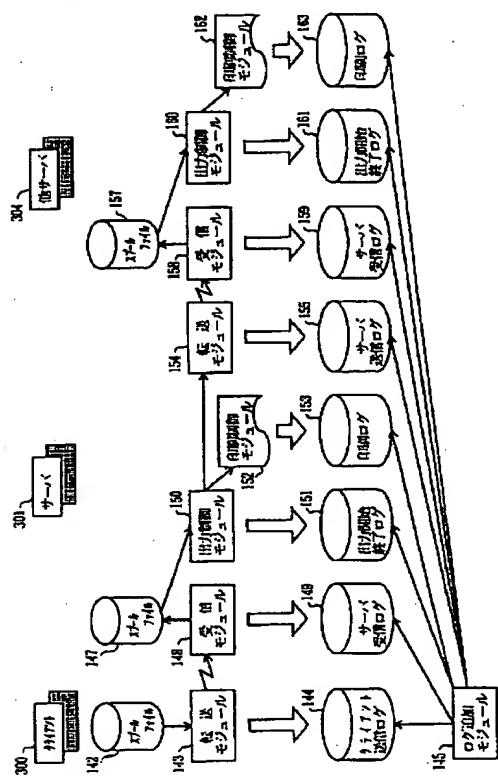
【図16】

## CAPCALプリントプロセッサの動作を示すフローチャート



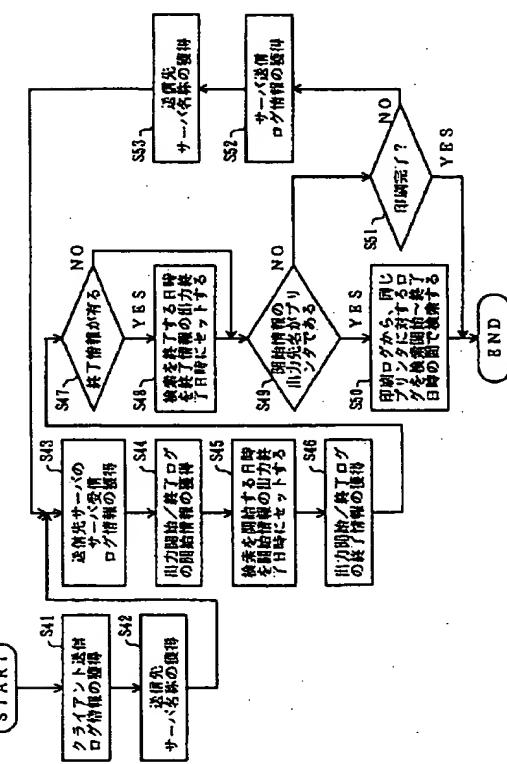
【四】

#### 本発明の一実施例に係るドキュメント出力監視処理部の構成を示すブロック図



【图25】

本発明の一実施例に係わるドキュメント出力監視方法を示すフローチャート



フロントページの続き

(72)発明者 新堂 幸博  
東京都羽村市栄町3丁目2番1号 カシオ  
計算機株式会社羽村技術センター内

(72) 発明者 波多野 英二  
東京都羽村市栄町 3 丁目 2 番 1 号 カシオ  
計算機株式会社羽村技術センター内

(72) 発明者 天利 忠義  
東京都羽村市栄町 3 丁目 2 番 1 号 カシオ  
計算機株式会社羽村技術センター内

**JPO and NCIP are not responsible for any  
damages caused by the use of this translation.**

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

## CLAIMS

---

### [Claim(s)]

[Claim 1] A drawing data generation means to generate drawing data, and the operating system which manages said drawing data generation means, A directions means to direct generation of the spool file by said drawing data to said operating system, The printing system characterized by having a printing conditioning means to set up the printing conditions of said drawing data, an output-control information generation means to generate the output-control information on said drawing data based on said printing conditions, and an output-control information storing means to store said output-control information.

[Claim 2] The printing system according to claim 1 characterized by having further the processor which generates the printer management file corresponding to said drawing data based on the output-control information generated by said output-control information generation means.

[Claim 3] A receiving means to receive the contents of said printer management file, and the contents of said spool file, The distribution place definition table which described the correspondence relation between a distribution place name and an output unit name, A retrieval condition acquisition means to acquire the retrieval conditions of a print job from the contents of said printer management file, A character string extract means to extract a character string from said print job based on said retrieval conditions, A comparison means to compare the character string extracted by said character string extract means with the distribution place name described by the distribution place definition table, The printing system according to claim 1 or 2 characterized by having an output means to output the print job from which the character string which is in agreement with said distribution place name was extracted to the output unit corresponding to said distribution place name.

[Claim 4] It is the printing system according to claim 3 which said printing conditioning means is equipped with a virtual form display means display an imagination form, a division means divide said imagination form, and a division block-definition means specify said divided rectangle field, and is characterized by for said character string extract means to extract a character string from the range of the print job corresponding to the rectangle field specified by said division block-definition means.

[Claim 5] A receiving means to receive the contents of said printer management file, and the contents of said spool file, The distribution place definition table which described the correspondence relation between a distribution place name and an output unit name, The group definition table which described the link information to a group name and said distribution place name, A distribution place information acquisition means to acquire distribution place information from the contents of said printer management file, When the assignment distribution mode of said distribution place information is group designation, the group name shown using said distribution place information and the linked distribution

place name are searched from said group definition table. The 1st output means which outputs a print job to the output unit corresponding to said distribution place name, When the assignment distribution mode of said distribution place information is logical name assignment, the distribution place name which is in agreement with the logical name shown using said distribution place information is searched from said distribution place definition table. The 2nd output means which outputs a print job to the output unit corresponding to said distribution place name, The printing system according to claim 1 or 2 characterized by having the 3rd output means which outputs a print job to the output unit which is in agreement with the physical name shown using said distribution place information when the assignment distribution mode of said distribution place information is physical name assignment.

[Claim 6] A transmitting result record means to record the transmitting result of a spool file with printing demand former information, A receiving result record means to record the receiving result of said spool file with said printing demand former information, An output record means to record the output of said spool file with said printing demand former information, A printing result record means to record the printing result of said spool file with said printing demand former information, When the output destination change of said spool file is a printer under jurisdiction of a self server, When it searches for the printing result recorded on said printing result record means and the output destination change of said spool file is a printer under jurisdiction of other servers, A printing system given in any 1 term of claims 1-5 characterized by having further a retrieval means to search for the transmitting result recorded on said transmitting result record means.

[Claim 7] The storage possible in computer reading stored the program which makes a computer perform the function which generates drawing data, the function realize an operating system, the function of making said operating system generating the spool file by said drawing data, the function of setting up the printing conditions of said drawing data, the function that generate the output-control information on said drawing data based on said printing conditions, and the function of storing said output-control information.

---

## DETAILED DESCRIPTION

---

### [Detailed Description of the Invention]

#### [0001]

[Field of the Invention] Especially this invention is applied to a client-server printing system about a printing system, and is suitable.

#### [0002]

[Description of the Prior Art] In recent years, the spread of Windows of Microsoft Corp. is remarkable as an operating system of the computer for office work or basic business.

[0003] Drawing 26 is a block diagram which explains the conventional printing approach at the time of using Windows as an operating system. In drawing 26 , when a user specifies printing to a printer driver 203 as application 201, application 201 executes a printing instruction to GDI (graphical device interface function)202. If the printing instruction from application 201 is executed, GDI202 will acquire required information from a printer driver 203, and will create a spool file (an EMF file and job description file) 204. The spool system 205 of Windows will start the De Dis Poole processing, if it

checks that the appointed port is in the condition which can be printed. By this De Dis Poole processing, printing control includes the print processor 207 through the port thread 206. And print data are outputted for print data to an airline printer 211 through the parallel port driver 210 over a port monitor 209 through the local print provider 208.

[0004] Here, as an operating system, when Windows is used, since Windows is a personal-oriented operating system, there is an unsuitable point in extensive printing in basic business. For example, it cannot print to two or more printers by one printing action at coincidence. For this reason, when performing distribution of data etc., copy printing starting is applied and it takes the time and effort which classifies printed matter manually and distributes it by mail etc. further. Moreover, e-mail cannot be substituted for printing of managed documents, such as total data, printing of a lot of documents, etc. by application operation although e-mail can be substituted for the small quantity and fixed document which were created with the word processor etc. by the spread of the mails of these days. Furthermore, when performing a lot of printing, it cannot print dispersedly to two or more printers, but it is necessary to repeat and specify an initiation page, a termination page, and an output destination change printer.

[0005] In order to solve these problems, in the former, API (application interface function) of dedication for basic operating document printing was prepared, and there was the approach of compensating the defect of the printing control which operating systems, such as Windows, have using the API.

[0006] Drawing 27 is a block diagram explaining the conventional distributed printing approach. In drawing 27 , application 221 calls API223 as a printing control function of dedication. Here, the interface which can receive two or more groups of a printer name, an initiation page, and a termination page makes distributed printing possible by preparing this interface in \*\*\*\*\* and API223 at API223. If API223 receives drawing data from application 221 as a parameter 222 while receiving two or more groups of a printer name, an initiation page, and a termination page, it stores in a spool file 226 the parameter 222 and drawing data which were received via an operating system 224 and the file driver 225.

[0007] On the other hand, the spool writer 227 is polling a certain decided directory (spool directory), and if the completed spool file 226 is detected, it will start read-out of a spool file 226. And while taking out the parameter 222 stored in the spool file 226, the drawing data of each page are read from a spool file 226 one by one. Then, GDI228 is called and drawing data are sent to the appointed printer 230 through a printer driver 229 according to the information on a parameter 222. Consequently, it becomes possible to perform printing of the appointed page by the appointed printer 230, and the distributed printing using two or more printers becomes possible by specifying two or more groups of a printer name, an initiation page, and a termination page.

[0008] In addition, GDI228 is the set of functions for graphical device drawing closely connected with the operating system 224.

[0009]

[Problem(s) to be Solved by the Invention] However, by the technique of offering conventional API223, it was inapplicable only to the application 221 of the dedication which can call API223. For this reason, since there was no function to call API223 in general-purpose package applications, such as a word processor, there was a problem of being inapplicable in general-purpose package applications, such as a word processor. Moreover, there was also a problem that the application 221 for printing had to be built

using offered API23.

[0010] Furthermore, there is a trouble of coming by the conventional printing approach from the defect of printing control of an operating system 224, and these troubles were not able to be solved by the approach of offering API23. For example, although the notice which tells that printing having been completed and an error occurred is shown in the screen on the computer to which the printer was connected, these things are not understood in the client machine which directed printing activation through the communication network. Moreover, re-printing which changed re-printing which respecified the page to the job under halt in the printer error, and an output printer cannot be performed. Furthermore, the printer on the computer connected by WAN (wide area network) cannot be specified directly, and it cannot print.

[0011] Moreover, there were the following as a trouble about the dependency to the output unit and application of an operating system 224. First, the printer kind and the printer manufacturer were specified and the print facility was offered, and in order to use the various printers of a network environment effectively, it took great time and effort. For this reason, the document created with application based on the spec. of specific output units (printer etc.) became a thing depending on the physical specification of an output unit, and did not have portability. Consequently, when outputted to the output unit with which physical specifications differ, there was a case where the result to expect could not be obtained.

[0012] Next, when aimed at the document outputted with all the applications on a system, after [ no ] the output were able to be carried out with application. For example, various services, such as an addition and modification of an output destination change, actuation in a page unit, and actuation of the order of an output, were impossible.

[0013] Moreover, there were the following as a trouble about the distribution printing method of an operating system 224. First, when a printout (document) was distributed, he directs the printer of a distribution place directly and always needed to be conscious of system environment (printer name to a distribution place). Moreover, specified all the names of the printer used as the candidate for distribution, the distribution place was specified for every document, and things had become a burden for the user smoothly.

[0014] Next, when determining a distribution place from the contents of a document, it is necessary to specify the field of the document for determining a distribution place but, and since all the character strings of the specified field are treated as a distribution place candidate, in order to direct a distribution place certainly, a limit occurs in the character string which can be used in the specified field. For this reason, when a certain distribution place was temporarily made into the outside of an object, complicated actuation of deleting a character string from a document, the time and effort of next restoration, etc. were needed, and there was a case where it became the factor of an operation mistake. Furthermore, since the degree of freedom is high, the implementation approach is complicated and a technical problem occurs for the engine performance etc.

[0015] Moreover, there were the following as a trouble about the block-definition method of an operating system 224. Since it was a setting method depending on the configuration (a paper size, how to place) of each document although it is necessary to include the function in each application when it is required for some the purpose to specify the field of a printing document, it needed to set up for every

document. Furthermore, even if set up once, there was time and effort of resetting whenever it changes the configuration of a document.

[0016] Moreover, there were the following as a trouble about the document output supervisory-control method of an operating system 224. Although it is management of printing processing (printer output) and the monitoring function which made the printer the subject was offered in the existing system, the monitor which made the subject printing processing which each user supplied was impossible, and the user was difficult to supervise simply all printing processings that he supplied.

[0017] Then, the 1st purpose of this invention is offering the printing system which can deal with the document drawn up with general-purpose application on a network. Moreover, the 2nd purpose of this invention is offering the printing system which can perform easily distribution printing of the document drawn up with general-purpose application.

[0018] Moreover, the 3rd purpose of this invention is offering the printing system which can supervise the operating state of the printer connected to the network by the client side.

[0019]

[Means for Solving the Problem] A drawing data generation means to generate drawing data according to this invention in order to solve the technical problem mentioned above, The operating system which manages said drawing data generation means, and a directions means to direct the generation of a spool file based on said drawing data to said operating system, It has a printing conditioning means to set up the printing conditions of said drawing data, an output-control information generation means to generate the output-control information on said drawing data based on said printing conditions, and an output-control information storing means to store said output-control information.

[0020] Since it becomes possible to make it spool with the operating system which manages that drawing data generation means for the drawing data generated by the drawing data generation means by this, if it is the application which operates on that operating system, any applications will become possible [ using it as a drawing data generation means ]. Moreover, since it becomes possible to store output-control information separately with the spool file based on drawing data by establishing an output-control information storing means to store output-control information, it becomes possible to add freely from after the output limiting facility with which an operating system was not equipped.

Consequently, in the conventional operating system, it becomes possible to perform notice processing of the document distribution processing by the network environment which was not able to be realized easily, distributed printing processing, distribution place automatic detection processing or a printer error, and the completion of printing etc. to the document drawn up with general-purpose application.

[0021] Moreover, if this invention is caused like 1 voice, based on the output-control information generated by said output-control information generation means, it has further the processor which generates the printer management file corresponding to said drawing data.

[0022] It becomes possible to perform processing and actuation of a printer management file by this in the original format which became independent of an operating system, and it becomes possible to perform printing processing of a document flexibly with various gestalten.

[0023] Moreover, the distribution place definition table which described the correspondence relation between a distribution place name and an output unit name when this invention was caused like 1 voice, A retrieval condition acquisition means to acquire the retrieval conditions of a print job from the

contents of the printer management file which received, A character string extract means to extract a character string from said print job based on said retrieval conditions, It has a comparison means to compare the character string extracted by said character string extract means with the distribution place name described by the distribution place definition table, and an output means to output the print job from which the character string which is in agreement with said distribution place name was extracted to the output unit corresponding to said distribution place name.

[0024] Based on the distribution place detected automatically, it becomes possible to specify an output unit by becoming possible to distinguish automatically the character string which is contained in a document based on the retrieval conditions sent from the client and which was specified beforehand by this, and referring to a distribution place definition table, while becoming possible to detect a distribution place automatically from a print job. Consequently, it becomes possible to print automatically to all the distribution places that correspond to retrieval conditions only by performing one printing directions, without becoming possible to save the time and effort which specifies the distribution place of a printing document, and being conscious of the printer name corresponding to a distribution place.

[0025] Moreover, if this invention is caused like 1 voice, he has a virtual form display means to display an imagination form, a division means to divide said imagination form, and a division block-definition means to specify said divided rectangle field, and is trying to extract a character string from the range of the print job corresponding to the rectangle field specified by said division block-definition means.

[0026] Since this enables it to specify a code area in the direction of the four directions of a form, middle, etc., when it is required to specify the field of a printing document because of retrieval of a distribution place etc., one field setup in an imagination form enables it to set up the code area of a distribution place appropriately, without being dependent on the configuration (a paper size, how placing) of each printing document.

[0027] Moreover, the distribution place definition table which described the correspondence relation between a distribution place name and an output unit name when this invention was caused like 1 voice, The group definition table which described the link information to a group name and said distribution place name, A distribution place information acquisition means to acquire distribution place information from the contents of the printer management file which received, When the assignment distribution mode of said distribution place information is group designation, the group name shown using said distribution place information and the linked distribution place name are searched from said group definition table. When the assignment distribution mode of said distribution place information is logical name assignment, the distribution place name which is in agreement with the logical name shown using said distribution place information is searched from said distribution place definition table. When the assignment distribution mode of said distribution place information is physical name assignment, he is trying to output a print job to the output unit which is in agreement with the physical name shown using said distribution place information.

[0028] Only by registering the correspondence relation between the distribution place name for N base, and an output unit, or becoming possible to define two or more output units as one logical distribution place, and specifying one distribution place by this Since it becomes possible to determine automatically two or more output units corresponding to the distribution place, while becoming possible

to specify a distribution place not by the physical output unit but by the logical name, it becomes possible to distribute a document to two or more output units by one assignment. For this reason, it becomes that it is possible in printing automatically to all the distribution places that correspond to conditions with one printing directions, without becoming possible to save the time and effort which makes the multiple selection of the distribution place to wish at the time of printing activation, specifies a distribution place for every printing document, or specifies all the names of the printer which is a candidate for distribution, and being conscious of the printer name corresponding to a distribution place. [0029] Moreover, a transmitting result record means to record the transmitting result of a spool file with printing demand former information if this invention is caused like 1 voice, A receiving result record means to record the receiving result of said spool file with said printing demand former information, An output record means to record the output of said spool file with said printing demand former information, A printing result record means to record the printing result of said spool file with said printing demand former information, When the output destination change of said spool file is a printer under jurisdiction of a self server, When it searches for the printing result recorded on said printing result record means and the output destination change of said spool file is a printer under jurisdiction of other servers, By this equipped with a retrieval means to search for the transmitting result recorded on said transmitting result record means [ when a transfer of the spool file on a network is performed ] The transmitting result, the receiving result, output, and printing result of the spool file By becoming possible about whether for it to have originated in which printing demand origin, and to have been obtained to judge easily, and investigating the contents of a transmitting result record means, a receiving result record means, an output record means, and the printing result record means It becomes possible to recognize how the printing demand of it was processed by the printer of which server. Consequently, when printing is performed in which server, a client becomes possible [ grasping a printing situation correctly ], while becoming possible [ performing a printing demand ], without being conscious of by which server the printer is managed.

[0030]

[Embodiment of the Invention] Hereafter, it explains, referring to a drawing about the printing system concerning one example of this invention. In addition, the following examples explain the output unit administration support (Following CAPCAL is called) control system under the operating system by Windows. Moreover, in this CAPCAL control system, the output document of all the applications on a system can be made into a processing object.

[0031] Drawing 1 is the block diagram showing the printing structure of a system concerning one example of this invention. In drawing 1 , a client 300 and a server 301 are connected through communication networks 302 and 303, and the server 301 is connected with the printer 9. Here, a communication network 302 is LAN (Local Area Network), and communication networks 303 are LAN, WAN (Wide Area Network), and the Internet. Application 1, GDI2, the Windows printing control section 3, and the client control section 4 are formed in a client 300, and the Windows printing control section 5, the server control section 6, GDI7, and the printer output-control section 8 are formed in the server 301. And when processing by the Windows printing control sections 3 and 5 is performed when performing printing processing by Windows, and performing CAPCAL control, processing by the client control section 4, the server control section 6, and the printer output-control section 8 is performed. In

addition, when choosing CAPCAL control, the CAPCAL driver prepared in CAPCAL control is specified with application 1, and GDI2 is called.

[0032] Drawing 2 is a block diagram explaining the configuration of the client control section 4 of drawing 1 . In drawing 2 , the CAPCAL driver 14, the CAPCAL driver property control thread 15, the output-control file 17, the job description file 18, the EMF files 19 and 24, the CAPCAL client definition file 20, the CAPCAL print processor 21, the CAPCAL job description file 23, the CAPCAL destination definition file 25, the CAPCAL client main process 26, and the CAPCAL job transfer thread 27 are formed in the client control section 4. Moreover, the spooler process 11, the local print provider 12, and the print processor 13 are formed in the Windows printing control section 3. Here, the client control section 4 incorporates the print data which the application 1 which works on the operating system by Windows drew by the processing environment of CAPCAL, and controls processing until it transmits to a server 301.

[0033] When processing the print data which application 1 drew in the CAPCAL environment, application 1 specifies the CAPCAL driver 14 as a virtual driver, and calls GDI2. GDI2 acquires a drawing function from the CAPCAL driver 14, changes it into the drawing function in which the CAPCAL driver 14 has the drawing primitive which application 1 required, and is spooled to the system spool queue 16. The job description file 18 which stored in this system spool queue 16 the user name which carried out the printing demand, the application name, etc., and the EMF file 19 which formatted and stored drawing information are created. EMF is Enhanced. It is the abbreviation for Meta-File and is the logical storing format of the drawing data in Windows. In addition, the format of the job description file 18 and the EMF file 19 is completely [ as the standard format for Windows ] the same. For this reason, the default function with which Windows is equipped can be made to generate the job description file 18 and the EMF file 19.

[0034] Moreover, the CAPCAL driver 14 is written in the system spool queue 16 by considering output-control information peculiar to CAPCAL as the output-control file 17. In addition, this output-control information can be set up by opening a dialog screen from the CAPCAL driver 14 side, and making an operator specify printing conditions, such as distribution and distribution. When the output-control information inputted by the operator is memorized for every document and the document is printed next time, you may make it use as a default the output-control information on the document inputted last time here.

[0035] Completion of a spool of all the drawing data of application 1 calls the CAPCAL print processor 21. The call of this CAPCAL print processor 21 is performed by specifying the CAPCAL driver 14 with application 1. In addition, when other drivers are specified with application 1, the print processor 13 with which Windows is equipped is called.

[0036] The CAPCAL print processor 21 incorporates the contents of the job description file 18 and the output-control file 17 from the system spool queue 16, and writes them in the CAPCAL spool queue 22 as a CAPCAL job description file 23. Here, the default of output-control information is defined as the CAPCAL client definition file 20, and when output-control information is not stored in the output-control file 17, output-control information is incorporated from the CAPCAL client definition file 20.

[0037] Moreover, the CAPCAL print processor 21 writes the EMF file 19 of the system spool queue 16

in the EMF file 24 of the CAPCAL spool queue 22 as it is. If the CAPCAL job description file 23 and the EMF file 24 are written in the CAPCAL spool queue 22, the CAPCAL print processor 21 will notify this to the CAPCAL job transfer thread 27.

[0038] The CAPCAL job transfer thread 27 transmits the CAPCAL job description file 23 and the EMF file 24 to a server 301 based on the transfer means defined beforehand. In addition, Server Name of a transfer means and the destination is stored in the CAPCAL destination definition file 25, and the CAPCAL job transfer thread 27 determines the server of a transfer means and the destination according to the contents of the CAPCAL destination definition file 25. In addition, the communication network 28 using the file transfer protocol (ftp) as a transfer means and the communication network 29 using the attached file of e-mail are usable. Moreover, transfer means other than these may be used.

[0039] Thus, the CAPCAL driver 14 as a virtual driver is formed in the client control section 4. And while making the job description file 18 and the EMF file 19 spool to the system spool queue 16 by GDI2, the CAPCAL driver 14 spools the output-control file 17 to the system spool queue 16 uniquely. By this, it becomes possible to make the format of the job description file 18 and the EMF file 19 in agreement with the standard format for Windows, and it becomes that it is possible to realize notice processing of document distribution processing, distributed printing processing, distribution place automatic detection processing, a printer error, and the completion of printing etc. about all the applications that operate on Windows containing a general-purpose software package, without changing no application interface of Windows. In order to realize notice processing of document distribution processing, distributed printing processing, distribution place automatic detection processing, a printer error, and the completion of printing etc., it becomes unnecessary consequently, to newly build the application for printing.

[0040] Drawing 3 is a block diagram explaining the configuration of the server control section of drawing 1 . In drawing 3 , the CAPCAL receiving thread 30 for e-mail, the CAPCAL receiving thread 31 for ftp, the CAPCAL server main process 32, the CAPCAL job description files 34 and 39, the EMF files 35 and 40, the CAPCAL server side definition file 36, the distribution distributed-processing thread 37, the output threads 41 and 42 classified by printer, and the CAPCAL job transfer thread 43 are formed in the server control section 6.

[0041] If the CAPCAL receiving threads 30 and 31 are started by the CAPCAL server main process 32, the CAPCAL receiving threads 30 and 31 will receive the CAPCAL job description file 23 and the EMF file 24 from the CAPCAL job transfer thread 27. And the contents of the CAPCAL job description file 23 which received, and the EMF file 24 are written in the CAPCAL job description file 34 and the EMF file 35 of the CAPCAL spool document queue 33. Here, the CAPCAL receiving threads 30 and 31 manage so that a file name (job ID) may not overlap. If the writing to the CAPCAL spool document queue 33 is completed, the CAPCAL server main process 32 will start the distribution distributed-processing thread 37, and will notify the write-in completion to the CAPCAL spool document queue 33 to the distribution distributed-processing thread 37.

[0042] The distribution distributed-processing thread 37 incorporates output-control information from the CAPCAL job description file 34. Moreover, the contents of the CAPCAL server side definition file 36 are read beforehand. And from the contents of the output-control information side and server side definition file, the distribution distributed-processing thread 37 determines an output printer and the

initiation page to output, and a termination page, and supplies a printing demand to the CAPCAL spool print queue 38 prepared every printer 9. The correspondence relation between a distribution place name and a printer name (or Server Name) is stored, and when the distribution place is specified by the logical name, an output printer can be determined as the CAPCAL server side definition file 36 by referring to the CAPCAL server side definition file 36. If the spool to the CAPCAL spool print queue 38 is completed, the printer 9 with a printing demand will be notified to the output thread 41 classified by printer taken charge of specially.

[0043] The output thread 41 classified by printer sends drawing data to the Windows printing control 5 by taking out the EMF file 40 from the CAPCAL spool print queue 38, and calling GDI7. In addition, when printers other than printer 9 which the server 301 has managed are specified as an output printer according to the contents of the output-control information side and CAPCAL server side definition file 36, the distribution distributed-processing thread 37 sends a printing demand to the server which has managed the printer specified through the communication network 44 from the CAPCAL job transfer thread 43. Thus, when the distribution place is specified by the logical name by storing the correspondence relation between a distribution place name and a printer name (or Server Name) in the CAPCAL server side definition file 36, it becomes possible to determine an output printer, and a user becomes possible [ performing distribution printing ], without being conscious of an output printer name.

[0044] Moreover, it becomes possible to print using various printers on a network, without becoming possible to obtain a suitable output and being conscious of a printer kind or a printer manufacturer, when the output threads 41 and 42 classified by printer are formed for every printer, the output threads 41 and 42 classified by printer perform an emulation and the specifications of a printer differ.

[0045] Drawing 4 is a block diagram explaining the configuration of the printer output-control section of drawing 1 . In drawing 4 , the SENTORO port monitor 48 and the printer output Management Department 50 are established in the printer output-control section 8, the CAPCAL server main process 32, the EMF file 40, and the output thread 41 classified by printer are formed in the server control section 6, the Raw spool file 46, the print processor 47, and the spooler process main 49 are formed in the Windows printing control section 5; and the CAPCAL client main process 26 is formed in the client 300.

[0046] The output thread 41 classified by printer sends drawing data to the Windows printing control section 5 by taking out the EMF file 40 from the CAPCAL spool print queue 38, and calling GDI7. GDI7 communicates with the printer driver 45 for printer 9 by which printing assignment was carried out, and changes the EMF file 40 into the command (ESC sequence) of the printer 9 dedication by which printing assignment was carried out. The changed ESC sequence is stored in the system spool queue of Windows. This is called Raw spool file 46. The Raw spool file 46 consists of only pure printer commands (ESC sequence). The Raw spool file 46 is read by the spooler process main 49 of Windows by the print processor 47, is a drawing data unit for 1 page, and is sent to the SENTORO port monitor 48.

[0047] When the interface of a printer 9 and a server 302 is LAN, it is sent to a LAN port monitor and, in the case of other interfaces, the port monitor for the interfaces is sent. In addition, the processing so far is the same as data processing by the Windows printing control section 5.

[0048] The SENTORO port monitor 48 outputs the command data of the printer 9 received by the spooler process main 49 course to cent ROPOTO. Here, when an error occurs to a printer 9 and command data are not able to be outputted, the SENTORO port monitor 48 notifies the printer output Management Department 50 of the job ID which identifies a printer name, an error code, and the original document. Moreover, the SENTORO port monitor 51 also notifies the printer output Management Department 50 of the job ID which identifies a printer name, an error code, and the original document, when an error occurs to the printer connected to the SENTORO port monitor 51. Here, Job ID is ID which specifies the print job which exists in the structure of Windows printing control, and the printer output Management Department 50 can acquire a document name, the original client host name which carried out the printing demand, and an original user name by notifying the printer output Management Department 50 of Job ID.

[0049] The printer output Management Department 50 will notify such information to the CAPCAL server main process 32, if a printer name, an error code, and Job ID are received from the SENTORO port monitor 48.

[0050] The CAPCAL server main process 32 detects the client host name which published the printing demand from the acquired job ID, and notifies a document name, a user name, and an error code to the CAPCAL client main process 26 which works on the client host. Moreover, such information is displayed by performing the screen call 52.

[0051] The CAPCAL client main process 26 will display a user name, the document name under printing, and an error message by performing the screen call 53, if a document name, a user name, and an error code are received from the CAPCAL server main process 32.

[0052] Moreover, the SENTORO port monitor 48 will notify the printer output Management Department 50 of a printer name and Job ID, if the output of one job is completed completely. The printer output Management Department 50 will notify such information to the CAPCAL server main process 32, if a printer name and Job ID are received from the SENTORO port monitor 48.

[0053] The CAPCAL server main process 32 detects the client host name which published the printing demand from the acquired job ID, and notifies a document name and a user name to the CAPCAL client main process 26 which works on the client host. Moreover, such information is displayed by performing the screen call 52.

[0054] The CAPCAL client main process 26 will display the message which shows that printing of a document was completed by performing the screen call 53, if a document name and a user name are received from the CAPCAL server main process 32.

[0055] Thus, since the SENTORO port monitor 48 becomes possible [ notifying a printing error and the completion of printing to the client 300 which became able / the printer output Management Department 50 / to recognize the client host name which carried out the printing demand, and carried out the printing demand ] by notifying the printer output Management Department 50 of the job ID which identifies the original document, it becomes that it is possible in recognizing the printing situation of a document at a client 300 side.

[0056] Hereafter, the printing system of drawing 1 is explained more to a detail. Drawing 5 is a block diagram explaining actuation of the virtual driver 14, and the flow of processing is shown until the CAPCAL spool file 57 is created.

[0057] In drawing 5 , when a user specifies printing to the virtual driver 14 as application 1, application 1 executes a printing instruction to GDI2 (in CAPCAL, it becomes spool printing automatically). GDI2 is device management information (Windows system \*\* value.), when a printing instruction is executed. Addition of information original with a printer driver is possible. following and DEVMODE information -- describing -- it requires of the virtual driver 14. The virtual driver 14 sets the common information (a paper size, form kind, etc.) independent of a manufacturer or a printer kind, and the name of the output-control file 17 as DEVMODE information, and returns them to GDI2. GDI2 is DEVMODE information and drawing information (GEIINFO information: printer driver drawing capacity.), when required information is received from the virtual driver 14. Information which a printer driver returns to GDI2. With a Windows system \*\* value, the EMF file 19 and the job description file 18 are created. Thus, in CAPCAL control printing, when GDI2 receives DEVMODE information from the virtual driver 14, a Windows system creates the job description file (Windows system \*\* value) 18.

[0058] On the other hand, the virtual driver 14 creates the output-control file 17 by the printer property (the information on assignment distribution, automatic distribution, and distributed printing is specified) of the virtual driver 14, and the file name added to DEVMODE information by the output-control information saved beforehand. The spool system 54 of Windows checks that the specified port is in the condition which can be printed, and starts the De Dis Poole processing. By the De Dis Poole processing, control includes the CAPCAL print processor 21 through the port thread 55. The CAPCAL print processor 21 acquires the name of the output-control file 17 which was added to DEVMODE information and added to DEVMODE information by the virtual driver 3, and the name of the EMF file 19 by reading the job description file 18 which the Windows system created. the name of the acquired output-control file 17, and the name of the EMF file 19 -- being based -- the job description file 18 -- and -- \*\* -- output-control information and print data are acquired from the EMF file 19, respectively, and the CAPCAL spool file 57 (the CAPCAL job description file 23 and EMF file 24) is created.

[0059] The CAPCAL print processor 7 will realize functions, such as assignment distribution printing, automatic distribution printing, and distributed printing, by sending the contents of the CAPCAL spool file 57 to a server 301 through the CAPCAL job transfer thread 27, if the CAPCAL spool file 57 is created.

[0060] Thus, while the virtual driver 14 makes GDI2 generate the job description file 18 and the EMF file 19 for the set-up DEVMODE information by delivery and GDI2, he is trying to pass the output-control information which stored output-control information in the output-control file 17 uniquely, and stored it in it uniquely to the CAPCAL print processor 21.

[0061] Consequently, since it becomes possible to make it spool with an operating system 2, if it is the application 1 which operates by that operating system 2 up one, the drawing data generated by application 1 While becoming possible to use any applications 1 By storing output-control information separately [ the job description file 18 and the EMF file 19 ] It becomes possible to add freely from after the output limiting facility with which an operating system 2 is not equipped. It becomes possible to perform notice processing of the document distribution processing by the network environment, distributed printing processing, distribution place automatic detection processing or a printer error, and the completion of printing etc. to the document drawn up with general-purpose application.

[0062] Drawing 6 is drawing showing the example of contents of the job description file 18. The job

description file 18 is equipped with the job description file header 61, the DEVMODE information 62, and the job description file footer 63 in drawing 6 . In addition, such information is Windows system \*\* values. The DEVMODE information 62 is created by the DEVMODE structure information on the virtual driver 14. Moreover, the printer driver private information 64 is added to the DEVMODE information 62 by the virtual driver 14. As DEVMODE information 62, there are a paper size, a form kind, page orientation, etc., and there is an output-control file name etc. as printer driver private information 64.

[0063] Drawing 7 is drawing showing the example of contents of the output-control file 17. In drawing 7 , the output-control file 17 was equipped with the CAPCAL output-control file header 71 and the output-control information 75, and the output-control information 75 is equipped with the mode-of-operation information 72, the document additional information 73, and the document attribute information 74. As for the CAPCAL output-control file header 71, version information etc. is stored including the information on the CAPCAL output-control file 17. As for the mode-of-operation information 72, the mode of operation (the normal mode / distribution mode / distributed mode) is stored. Moreover, the distribution place information specified by the virtual driver (CAPCAL driver) 14 is included. Distribution mode (automatic distribution / automatic distribution), the name of a distribution place, a retrieval page, search mode (break mode / qualification mode / attribute mode / size mode / color mode / rectangle mode), The class (an underline/half tone dot meshing) of qualification in qualification mode, the attribute class (a criterion / italic / bold letter / bold letter italic) in attribute mode, the character size in size mode, the color code in color mode, the field in rectangle mode, etc. are stored.

[0064] A form addition flag, the banner addition flag, the location of a form file, the form file name, the banner greeting sentence, etc. are stored including the additional information of the document with which the document additional information 73 is specified by the virtual driver 14.

[0065] A password flag, output mode (do not delay[ \*\*\*\*\* / time assignment printing / time-amount assignment printing / ]-print, / it does not print), the assignment time of time assignment printing, the assignment time amount of time-amount assignment printing, preservation mode (do not unconditionedness[ time assignment preservation / time-amount assignment preservation / ]-save, / it does not save), the assignment time amount of time assignment preservation, etc. are stored including the attribute information on a document that document attribute information 74 is specified by the virtual driver 14.

[0066] Drawing 8 is a flow chart which shows actuation of the virtual driver 14, and shows the processing which creates the job description file 18 and the output-control file 17 by the virtual driver 14. In this processing, the phase (steps S1-S3) when a user operates a CAPCAL printer driver property, and the printing phase (step S4- S6) of the virtual driver 14 are prepared.

[0067] In drawing 8 , a CAPCAL printer driver property is first opened like a general Windows printer driver (step S1). In addition, a CAPCAL printer driver property can be opened from a print folder or application.

[0068] Next, if a CAPCAL printer driver property is opened, the output-control information 75 is set up according to the display of a driver output-control setting screen, and the DEVMODE information 62 and the output-control information 75 are saved at registry (step S2). And from application 1, a printing

dialog is opened and printing is performed (step S3).

[0069] Next, the virtual driver 14 acquires the DEVMODE information 62 and the output-control information 75 from a CAPCAL printer driver property (step S4). In addition, if the DEVMODE information 62 and the output-control information 75 are not saved at a CAPCAL printer driver property, the value set as the virtual driver 14 is used.

[0070] Next, the virtual driver 14 passes the DEVMODE information (an output-control file name is included) 62 acquired from the CAPCAL printer driver property to GDI2 (step S5).

[0071] Next, the virtual driver 14 writes the output-control information 75 acquired from the CAPCAL printer driver property in the output-control file 17 by the output-control file name to which it is added by the DEVMODE information 62 (step S6).

[0072] Thus, when the virtual driver 14 returns the common information (DEVMODE information 62) independent of a printer kind or a manufacturer to a Windows system (GDI2) and makes GDI2 create the job description file 18, he is trying to give information (a form kind, a feed method, page orientation, etc.) required for printing. Moreover, the virtual driver 14 created the output-control file 17 uniquely, and has given the output-control information 75 required for distribution distributed processing to the CAPCAL printer processor 21. For this reason, it becomes possible to realize distribution printing processing independent of a manufacturer or a printer kind about the document drawn up with the general-purpose application 1. That is, a user can use a variety of printers 9 on a network, without that what is necessary is just to perform printing by the virtual driver 14 being conscious of the function specified by the printer kind or the manufacturer using the information which the virtual driver 14 gives, when making every printer on a network print.

[0073] Furthermore, the document created with application 1 is not dependent on the physical specification of an output unit (printer 9), and since it becomes possible [ obtaining the result expected even if it outputs to the printer by which specifications differ ], it can raise portability. Consequently, it becomes possible by making the output document of all the applications 1 on a system into a processing object to offer the logical device drive method independent of each specification of a physical output unit. Moreover, also when the functions of an output unit run short, the output expected by suitable emulation can be obtained.

[0074] Drawing 9 is drawing showing the setting screen of DEVMODE information. When a CAPCAL printer driver property is opened, by displaying the setting screen of this DEVMODE information, a feed location, a paper size, a print sheet, expansion/contraction, page orientation, a paper type, copy number of sheets, etc. can be inputted, and the DEVMODE information 62 on drawing 6 can be set up.

[0075] Drawing 10 is drawing showing the setting screen of automatic distribution information. When a CAPCAL printer driver property is opened, by displaying the setting screen of this automatic distribution information, assignment, search mode (break mode / qualification mode / size mode / color mode / attribute mode), the retrieval range, etc. can be inputted at the time of a retrieval page and printing, and the output-control information 75 on drawing 7 can be set up. Automatic distribution distributes by extracting a distribution place automatically out of the document distributed, even if a user does not specify a distribution place. Here, a retrieval page shows the page of the document which extracts a distribution place. In break mode, if it is considered that the field across which it faced by the specified character string is the retrieval range, for example, a parenthesis is specified, the character

string inserted in the parenthesis will serve as a distribution place. The character string embellished in qualification mode is detected as a distribution place, and it is considered that the character string which has the specified magnitude in size mode is a distribution place, and in color mode, it is considered that the character string which has the specified color is a distribution place, and it is considered in attribute mode. [ the character string of the specified attribute ] [ a distribution place ]

[0076] Moreover, the input of the retrieval range pinpoints the retrieval range by considering that this retrieval range is one sheet of imagination form, and specifying the field which divided the form and was obtained. For example, it is specified as the field of the upper right at the time of quadrisecting a virtual form by the natural centerline. And when you change assignment of the field in this virtual form into an actual form, let the field of the upper right which quadrisected the actual form, and was quadrisected and obtained by the natural centerline be the retrieval range. In addition, the approaches of division of a form may be eight division, 16 division, etc. besides quadrisection. By the input of this retrieval range, by making the output document of all the applications on a system into a processing object, while assignment of the field of a document is attained by the processing which became independent of application, the field of a document can be specified, without being dependent on the configuration (how placing a paper size and a form) of a document.

[0077] Drawing 11 (a) is a block diagram explaining the block-definition control section concerning one example of this invention. In drawing 11 (a), the block-definition control section 82 is formed in the user interface 81 of the virtual driver 14. The block-definition control section 82 incorporates the field information specified by a user by the virtual driver 14, and registers it as printed information of a document.

[0078] Drawing 11 (b) is drawing which explains how to carry out a block definition to a user by displaying the rectangle showing an actual form. In drawing 11 (b), the rectangle showing the portrait form 83 and the rectangle showing the form 85 of a landscape are displayed depending on how to place a form. And a user specifies the fields 84 and 86 which show the retrieval range to each of the portrait form 83 and the form 85 of a landscape.

[0079] Drawing 11 (c) is drawing which explains how to carry out a block definition to a user by displaying the rectangle showing the virtual form 87. In drawing 11 (c), the rectangle showing a form is set to MMI (man machine interface) as a square virtual form 87. A user specifies the rectangle field 88 on this virtual form 87.

[0080] And when performing printing, the information specified on the virtual form 87 is incorporated, and the rectangle field 88 on the virtual form 87 is changed about each case of the portrait form 89 and the form 91 of a landscape according to an actual form. When making the magnitude of the virtual form 87 equivalent to the magnitude of an actual form and changing it, it is made to correspond to the rate of variable power of the virtual form 87 in every direction, and the rectangle field 88 specified as the virtual form 87 is changed. It can change into the actual form 89 and the fields 90 and 92 which show the retrieval range on 91 by this, with the relative location of the rectangle field 88 on the virtual form 87 saved.

[0081] For example, in the case of which [ of the portrait form 89 and the form 91 of a landscape ], a distribution place is filled in at the upper column of forms 89 and 91 in many cases. For this reason, by specifying the field of the upper part on the virtual form 87 as a rectangle field 88, when the virtual

form 87 is changed into any of the portrait form 89 and the form 91 of a landscape, it is changed so that the fields 90 and 92 which show the retrieval range may be located in the upper column of forms 89 and 91, and it becomes possible to specify the retrieval range efficiently.

[0082] Thus, a field setup can be performed to all output documents, without including a block-definition function in application, when it is required to specify the field of a printing document for retrieval of a distribution place etc. by having offered the block-definition function on the hierarchy who became independent of application and a document, and having used MMI (man machine interface) of a block definition as the virtual form 87 of the square which became independent of the configuration of a document. This field setup is applicable to all the documents outputted from all applications. Moreover, a field setup has been independent of a document and all output documents can be targetted for it by one field setup. Furthermore, since it can set up without being dependent on the configuration (a paper size, how placing) of each document, while being able to aim at mitigation of a paper size and the block-definition activity by the difference in a way to place, once it sets up, resetting can be made unnecessary even if it changes the configuration of a document.

[0083] Drawing 12 is drawing showing the setting screen of assignment distribution information. When a CAPCAL printer driver property is opened, by displaying the setting screen of this assignment distribution information, distribution place mode (distribution place name assignment / group name assignment / printer name assignment), a distribution place name, a group name, a printer name, etc. can be inputted, and the output-control information 75 on drawing 7 can be set up. For example, when distribution place name assignment is chosen, it becomes possible to specify the distribution place of printed matter by logical names, such as the "Tokyo administration division", and it becomes possible to specify a distribution place, without being conscious of a printer name. Moreover, when group name assignment is chosen, it becomes possible to choose two or more distribution places by one assignment, and it becomes possible to mitigate the time and effort of selection of a distribution place. Furthermore, it becomes possible to add the selected distribution place freely on a screen, or to delete it, and it becomes possible to also make a change of a distribution place easily.

[0084] Drawing 13 is drawing showing the setting screen of the distributed printed information concerning one example of this invention. When a CAPCAL printer driver property is opened, by displaying the setting screen of this distributed printed information, the error of distinction of the distributed approach (/classification classified by number equal / number of copies / manual assignment), a color, and monochrome, the number of queings, and a printer etc. can be inputted, and the drawing 7 output-control information 75 can be set up. By specifying the distributed approach, it becomes possible to assign printing number of copies equally to two or more printers, or to assign it to them according to the printing capacity and system operating status of a printer. Moreover, by specifying distinction of a color and monochrome, only the page of a color is made to output to a color printer, or it becomes possible at it to make only the page of monochrome output to a monochrome printer. The printer by which the waiting for an output is saved can be avoided, priority can be given to a vacant printer, and it can be made to output by specifying the number of queings. When the distribution distribution thread 37 determines a printer by specifying the error of a printer, it can avoid treating the printer which the error has generated as a printer for distribution.

[0085] Drawing 14 is the block diagram showing the configuration of the CAPCAL print processor 21.

In drawing 14 , the print data read through the print-data reading control section 101 are stored in the print-data storing section 102. A data partition is carried out for every page in the print-data configuration section 103, and the print data stored in the print-data storing section 102 are sent to the CAPCAL spool file generation section 104. The CAPCAL spool file generation section 104 stores the page management information by the output for every page of print data in the page management information storing section 105, and outputs the print data of each page to the EMF file 24 of the CAPCAL spool file 57. If all print data are outputted to the EMF file 24 of the CAPCAL spool file 57, the CAPCAL job description file 23 will be created.

[0086] When creating the CAPCAL job description file 23, DEVMODE information and Windows job information are read from the print job information reading control section 106, CAPCAL output-control information is read from the CAPCAL output-control file reading control section 107, and CAPCAL User Information is read from the CAPCAL User Information reading control section 108. And DEVMODE information, Windows job information, CAPCAL output-control information, and CAPCAL User Information are stored in the CAPCAL job information storing section 109. Based on the information stored in the CAPCAL job information storing section 109, the print job information configuration section 110 constitutes print job information, and outputs it to the CAPCAL job description file 23 of the CAPCAL spool file 57 through the CAPCAL spool file generation section 105. Under the present circumstances, the page management information stored in the page management information storing section 105 is added to the print job information which consisted of the print job information configuration sections 110, and the CAPCAL spool file generation section 105 outputs it to the CAPCAL job description file 23.

[0087] Drawing 15 is drawing showing the example of contents of the CAPCAL job description file 23. The CAPCAL job description file 23 is equipped with the CAPCAL job description file header 111, the Windows job information 112, the DEVMODE information 113, the CAPCAL output-control information 114, CAPCAL User Information 115, and the print-data page management information 116 in drawing 15 .

[0088] The CAPCAL job description file header 111 includes information, such as a version number, as header information of the CAPCAL job description file 23. The Windows job information 112 is information acquired as job information on Windows, and includes information, such as a user name which requested printing, the total number of pages of a print job, a host name which the print job generated, a print job title name, and print job generating time of day.

[0089] The DEVMODE information 113 is the DEVMODE information 113 on the print job specified by application 1 and the virtual driver 14. The CAPCAL output-control information 114 has the same contents as the output-control information 75 on drawing 7 .

[0090] CAPCAL User Information 115 is a user's information defined for CAPCAL, and includes the information on a name, a firm name, a section signature, the telephone number, a FAX number, a password, etc.

[0091] The print-data page management information 116 is index information which shows the head location of each page on the file of print data. By referring to this print-data page management information 116, distribution printing and distributed printing can be performed easily.

[0092] Drawing 16 is a flow chart which shows actuation of the CAPCAL print processor 21. In

drawing 16 , the CAPCAL print processor 21 creates the EMF file 24 first (steps S11-S14). Specifically, the print data read through the print-data reading control section 101 are stored in the print-data storing section 102 (step S11). Next, from the print-data storing section 102, the print-data configuration section 103 acquires the print data for 1 page for every drawing unit, and outputs the print data of each page to the CAPCAL spool file 57 through the CAPCAL spool file generation section 104 (step S12). Under the present circumstances, the print-data page management information 116 is stored in the page management information storing section 105 of the CAPCAL spool file generation section 104. Next, when it judges whether there are any unsettled print data (step S13) and there are unsettled print data, it returns to step S12 and processing is repeated. Next, when it judges whether there are any spool data of non-reading (step S14) and there are spool data of non-reading, it returns to step S11 and processing is repeated.

[0093] The CAPCAL print processor 21 performs creation of the CAPCAL job description file 23, after outputting print data to the CAPCAL spool file 57 (steps S15-S17). Specifically the Windows job information 112 and the DEVMODE information 113 are read by the print job information reading control section 106, the CAPCAL output-control information 114 is read by the CAPCAL output-control file reading control section 107, and CAPCAL User Information 115 is read by the CAPCAL User Information reading control section 108. And the Windows job information 112, the DEVMODE information 113, the CAPCAL output-control information 114, and CAPCAL User Information 115 are stored in the CAPCAL job information storing section 109 as information for CAPCAL job description file 23 generation (step S15).

[0094] Next, the print-data page management information 116 stored in the page management information storing section 105 by the writing for every page of print data is written in the CAPCAL spool file 57 (step S16).

[0095] Next, the CAPCAL job information configuration section 110 writes the Windows job information 112 stored in the CAPCAL job information storing section 109, the DEVMODE information 113, the CAPCAL output-control information 114, and CAPCAL User Information 115 in the CAPCAL spool file 57 through the CAPCAL spool file generation section 104.

[0096] Thus, offer of completely different service from printing control of a Windows system of the CAPCAL print processor 21 is attained by spooling uniquely as print data of CAPCAL control, without carrying out the printout of the print data (spool file) required of printing control of a Windows system to a printer 9. Consequently, it becomes possible to provide a user with various services (actuation of the order of actuation/output of an output destination change etc.) per the page unit of a document, or document.

[0097] Here, since it is carrying out through printing control of a Windows system about creation of print data, it becomes that it is possible in providing various services, such as an addition and modification of an output destination change, actuation in a page unit, and actuation of the order of an output, also about printing from general-purpose application by making the output document of all the applications on a system into a processing object since it becomes possible to offer service of original printing control.

[0098] Next, the distribution printing processing concerning the example of this invention is explained. The distribution distributed-processing thread 37 of drawing 3 performs distribution printing processing.

In the distribution distributed-processing thread 37, distribution printing, distributed printing, and each actuation of usually printing are performed according to the contents directed for the output-control information 75 in the CAPCAL job description file 23. There are two kinds of distribution printings, automatic distribution and assignment distribution. Automatic distribution is the mode in which the contents of the print job determine a distribution place automatically. Assignment distribution is the mode in which a user specifies the distribution place of a print job beforehand. It can be defined in the CAPCAL job description file 23 which actuation is performed between these two modes.

[0099] Drawing 17 is a block diagram explaining the automatic distribution processing concerning one example of this invention, and shows an outline until it obtains the distribution place list in automatic distribution printing. In drawing 17, a server's 301 reception of the contents of the CAPCAL job description file 23 sent from the client 300 and the EMF file 24 stores these contents in the CAPCAL job description file 34 and the EMF file 35 of the CAPCAL spool document queue 33. The distribution distributed-processing thread 37 reads the contents of the CAPCAL job description file 34, and acquires the retrieval conditions 124 for searching the character string 122 as a distribution place candidate from a print job 121 from the CAPCAL job description file 34. There are a retrieval page, search mode (delimiter inside of /qualification / attribute / size / color / rectangle field), etc. in this retrieval condition 124, and the approach of extracting a distribution place candidate from a print job 121 can be specified as it by specifying this retrieval condition 124. In addition, this retrieval condition 124 can be set up by opening a CAPCAL printer driver property and displaying the setting screen of the automatic distribution information on drawing 10 on a client 300 side.

[0100] The distribution distributed-processing thread 37 searches a print job 121 based on this retrieval condition 124, and it acquires a distribution place definition from registry 123 while it acquires the character string 122 which suited the retrieval conditions 124 as a distribution place name. The correspondence relation between N base distribution cloth point name and an output unit name is stored in registry 123 as a distribution place definition. For this reason, in a comparator 125, the distribution place name acquired from the print job 121 is compared with the distribution place name defined as registry 123, and all the output units that should be distributed can be automatically determined by choosing only the character string 122 which is in agreement with the distribution place name defined as registry 123.

[0101] Drawing 18 is drawing showing the example of a retrieval condition at the time of automatic distribution. In drawing 18, there are a "delimiter", "qualification", a "attribute", "size", a "color", "inside of a rectangle field", etc. in search mode. A "delimiter" extracts the character string 122 between the start edge character strings and final-character trains which were specified as a distribution place candidate. In addition, a delimiter can be made to be able to output as it is, or can be made to replace and output in a tooth space by assignment at the time of printing. "Qualification" extracts the character string 122 embellished by an underline, half tone dot meshing, etc. as a distribution place candidate. In addition, a qualification alphabetic character can be made to be able to output as it is, or qualification can be made to cancel and output by assignment at the time of printing. A "attribute" extracts the character string 122 which has attributes, such as a criterion, italic, a bold letter, and bold letter italic, as a distribution place candidate. In addition, the alphabetic character in which the attribute was given can be made to be able to output as it is, or an attribute can be made to cancel and output by assignment at

the time of printing. "Size" extracts the character string 122 which has the specified character size (point size) as a distribution place candidate. A "color" extracts the character string 122 which has the specified color code as a distribution place candidate. In addition, the character string which has the specified color code can be made to be able to output in a color as it is, or can be made to change and output to black by assignment at the time of printing. "The inside of a rectangle field" extracts the character string 122 in the rectangle specified as a code area as a distribution place candidate.

[0102] Drawing 19 is a flow chart which shows actuation of the automatic distribution processing concerning one example of this invention, and shows the outline as which the distribution distribution thread 37 determines an output unit name. In drawing 1919 , the distribution distribution thread 37 acquires the retrieval conditions 124 from the output-control information 114 on the CAPCAL job description file 34 first. It is a retrieval page, search mode, and the condition detail that are acquired as retrieval conditions 124 (step S21). A retrieval page is a page of the print job 121 which searches. Search mode specifies which data in a retrieval page are made applicable to retrieval. As a condition detail is shown in drawing 18 , the information which changes with specified retrieval conditions 124 is stored.

[0103] Next, based on the acquired retrieval conditions 124, a print job 121 is searched, all the character strings 122 applicable to the retrieval conditions 124 are acquired from a print job 121, and it considers as the candidate of a distribution place (step S22). Next, a distribution place definition (all distribution places) is acquired from registry 123, and a distribution place definition information list is created (step S23). Only the part of the output destination change where the information from which the logical distribution place name and the physical output unit name corresponding to it became a pair serves as a candidate for distribution is registered into the distribution place definition. Next, only one distribution place definition is taken out out of the created distribution place definition information list (step S24).

[0104] Next, the distribution place candidate who acquired from the print job 121 is compared with the distribution place name acquired from the distribution place definition (step S25), and when the distribution place candidate who acquired from the print job 121, and the distribution place name acquired from the distribution place definition are in agreement, a corresponding output unit name is added to an output destination change list table (step S26).

[0105] Next, if there is a distribution place definition which is not compared with a distribution place definition information list (step S27), a new distribution place definition will be taken out from return and a distribution place definition information list to step S24, and comparison processing will be performed. and the case where all distribution place definitions in a distribution place definition information list are compared -- the distribution first thing to do -- a law -- processing is ended. After distribution place decision processing is completed, based on the created output destination change list, a print job 121 is supplied to each output destination change, and distribution printing is performed.

[0106] Thus, the retrieval conditions 124 of the distribution place specified by the client 300 side By transmitting as contents of the CAPCAL job description file 34 by the side of a server 301, and distinguishing automatically the character string which is contained in a document and which was specified beforehand based on the retrieval conditions 124 sent from the client 300 It becomes possible to detect a distribution place automatically from a print job 121 for the output document of all the applications 1 on a system. Consequently, it becomes possible to save the time and effort which

specifies a distribution place for every printing document, and it becomes possible about system environment to print automatically to all the distribution places that correspond to the retrieval conditions 124 with one printing directions, without being conscious of (the printer name corresponding to a distribution place). Furthermore, it becomes possible to give width of face to selection of a user's distribution approach, and it becomes possible to realize high distribution processing of a degree of freedom.

[0107] Drawing 20 is a block diagram explaining the assignment distribution processing concerning one example of this invention, and shows an outline until it obtains the distribution place in assignment distribution printing. In drawing 20 , a server's 301 reception of the contents of the CAPCAL job description file 23 sent from the client 300 and the EMF file 24 stores these contents in the CAPCAL job description file 34 and the EMF file 35 of the CAPCAL spool document queue 33. The distribution distributed-processing thread 37 reads the contents of the CAPCAL job description file 34, and acquires the distribution place information 131 from here. The distribution place information 131 is equipped with assignment distribution mode and the distribution place name of N individual, assignment distribution mode has group designation, logical name assignment (distribution place name assignment), and physical name assignment (printer name assignment), and there are a group name, a logical name (printer name), and a physical name (distribution place name) in a distribution place name. In addition, this distribution place information 131 can be set up by opening a CAPCAL printer driver property and displaying the setting screen of the assignment distribution information on drawing 11 on a client 300 side.

[0108] Here, the distribution distributed-processing thread 37 determines the output unit which should be distributed from the physical name of the distribution place name of the distribution place information 131, when the assignment distribution mode of the distribution place information 131 is physical name assignment.

[0109] On the other hand, when the assignment distribution mode of the distribution place information 131 is a group name or a logical name, the distribution distributed-processing thread 37 acquires distribution place definition information and group definition information from registry 132. As a distribution place definition, while the correspondence relation between the distribution place name for N base and an output unit name is stored, the link information to a group name and a distribution place is stored in registry 132 as a group definition. For this reason, the distribution distributed-processing thread 37 can determine the output unit which should be distributed based on the information defined as registry 132.

[0110] Drawing 21 is a flow chart which shows actuation of the assignment distribution processing concerning one example of this invention, and shows an outline until the distribution distribution thread 37 determines an output unit name in assignment distribution.

[0111] In drawing 21 , the distribution distributed-processing thread 37 acquires assignment distribution mode and a distribution place name from the output-control information 114 on the CAPCAL job description file 34 as distribution place information 131 (step S31).

[0112] Next, the distribution distributed-processing thread 37 acquires distribution place definition information and group definition information from registry 132 (step S32). The correspondence relation between the distribution place name for N base and an output unit name is registered into the

distribution place definition. The link to the distribution place definition corresponding to a logical group name and logical it is registered into the group definition.

[0113] Next, when the assignment distribution mode acquired from the distribution place information 131 is "physical name assignment" (step S33), the distribution place name shown using the distribution place information 131 turns into an output destination change name as it is. For this reason, an output destination change list is created as it is by making the distribution place name of the distribution place information 131 into an output destination change (step S34). On the other hand, when the assignment distribution mode acquired from the distribution place information 131 is "logical name assignment" (step S35), the distribution place name which \*\*\*\*\*\*(ed) from the distribution place information 131 is a logic name. For this reason, by referring to the distribution place definition information acquired from registry 132, an output unit is determined and an output destination change list is created (step S36).

[0114] Moreover, when the assignment distribution mode acquired from the distribution place information 131 is "group designation", the distribution place name acquired from the distribution place information 131 is a group name. For this reason, by referring to the group definition information and distribution place definition information which were acquired from registry 132, an output unit is determined and an output destination change list is created (step S37). After creation of an output destination change list is completed, a print job is supplied to each output destination change based on the created output destination change list, and distribution printing is performed.

[0115] Thus, it becomes possible to specify a distribution output destination change as arbitration per document for the output document of all the applications 1 on a system by transmitting the distribution place information 131 specified by the client 300 side as contents of the CAPCAL job description file 34 by the side of a server 301, and determining a distribution place based on the distribution place information 131 sent from the client 300.

[0116] Moreover, it becomes possible to define two or more output units at \*\* and one logical distribution place registering the correspondence relation between the distribution place name for N base, and an output unit into registry 123 and 132, and while becoming possible to distribute a document to two or more output units only by determining one distribution place, assignment becomes possible not by the physical output unit but by the logical name about a distribution place. For this reason, it becomes that it is possible in printing automatically to all the distribution places that correspond to conditions with one printing directions, without becoming possible to save the time and effort which makes the multiple selection of the distribution place to wish at the time of printing activation, specifies a distribution place for every printing document, or specifies all the names of the printer which is a candidate for distribution, and being conscious of the printer name corresponding to a distribution place.

[0117] moreover, a user -- free -- the distribution first thing to do -- a law -- it becomes possible to choose a method and it becomes possible by analyzing the contents of the document to determine a distribution place automatically from printing document data, or to specify a distribution place to a document by specifying directly the character string made into a distribution place object.

[0118] Furthermore, the positive distribution place assignment by the contents of a document is attained, and it becomes possible to make temporary change (distribution halt etc.) of a distribution place to ease and insurance. Consequently, high distribution processing of the degree of freedom which has width of

face in the distribution approach selection by the user is attained, and the system construction and employment of a degree of freedom which can respond to an employment situation flexibly [ it is high and ] are attained.

[0119] Next, document output monitor mode of processing concerning the example of this invention is explained. Drawing 22 is the block diagram showing the configuration of the document output monitor processing section concerning one example of this invention. In drawing 22 , a spool file 142 is generated by the client 300 by having performed printing. And the contents of the spool file 142 generated by the client 300 are transmitted to a server 301 through the transfer module 143. Under the present circumstances, the transfer module 143 outputs having transmitted the contents of the spool file 142 to the server 301 to the client transmitting log 144.

[0120] A server 301 will generate a spool file 147 on a server 301, if the contents of the spool file 142 transmitted from the transfer module 143 are received through the receiving module 148. Under the present circumstances, the receiving module 148 outputs having generated the spool file 147 on the server 301 to the server receiving log 149.

[0121] When a spool file 147 is generated on a server 301, the output-control module 150 writes out starting an output to the output initiation termination log 151, and makes the printing control module 152 start printing by the printer. If printing by the printer is completed, the printing control module 152 will output that printing by the printer was completed to the printing log 153, and will notify the completion of printing to the output-control module 150. The output-control module 150 which received the notice writes out that the output was completed to the output initiation termination log 151.

[0122] In the usual printing, printing to the printer which went via the server 301 from the client 300 is completed by processing mentioned above. On the other hand, when the printer specified from the client 300 is a thing to the other servers 304, a transfer of the contents of the spool file 147 is performed from a server 301 to the other servers 304.

[0123] When the printer which prints is under jurisdiction of the other servers 304, the output-control module 150 of a server 301 transmits the contents of the spool file 147 to the other servers 304 through the transfer module 154. Under the present circumstances, the transfer module 154 outputs having transmitted the contents of the spool file 147 to the other servers 304 to the server transmitting log 155.

[0124] The receiving module 158 of the other servers 304 will generate a spool file 157 on the other servers 304, if the contents of the spool file 147 transmitted from the transfer module 154 of a server 301 are received. Under the present circumstances, the receiving module 158 outputs having received the contents of the spool file 147 from the server 301 to the server receiving log 159.

[0125] When a spool file 157 is generated, the output-control module 160 of the other servers 304 writes out starting an output to the output open termination log 161, and makes the printing control module 162 start printing by the printer. If printing by the printer is completed, the printing control module 162 will output that printing by the printer was completed to the printing log 163, and will notify the completion of printing to the output-control module 160. The output-control module 160 which received the notice writes out 161 that the output was completed to an output initiation termination log.

[0126] The process to the transfer to a server 301 from a client 300 and printing serves as a transfer to the other servers 304 from a server 301, and the same control as the process to printing. Moreover, for

example, the transfer between the servers connected to WAN etc. transmits a spool file between the machines which can be transmitted and received and which are restricted, when printing towards the printer of a head office from a branch, and the transfer to the server which can be printed to the appointed printer from there may be performed. Thus, when the transfer to another server from the other servers 304 occurs further, transfer processing mentioned above is performed repeatedly.

[0127] In processing of drawing 22 , it means that the log which shows the processing process (result) of the spool files 142, 147, and 157 in the server 304 besides client 300 -> server 301-> was outputted to the client transmitting log 144, the server receiving log 149, the output initiation termination log 151, the printing log 153, the server transmitting log 155, the server receiving log 159, the output initiation termination log 161, and the printing log 163. The log trace module 145 becomes possible [ following all the processes from initiation of printing between different machines by relating with information / as opposed to / acquire information in an order from / these / each logs of these, and / one printing activation / to termination ].

[0128] Thus, the progress to the printing initiation by the client 300 and the printing activation by servers 301 and 304 and a printing activation result are stored in a log, respectively, and processing of the information display of two or more logs, actuation, etc. in which those information was stored is enabled by using as a key document output processing which a user can recognize. It becomes possible to supervise not an output unit but document output processing to a subject (result house keeping, monitor) about two or more document output processing which receives two or more output units which the user threw in by this in the system by which two or more servers which control an output unit exist on a network. Consequently, a printer is changed at the time of error generating, or a user becomes possible [ simplicity, insurance, and carrying out exactly ] about the directions to which re-printing which specifies a page is made to carry out while becoming possible [ supervising the printing processing which he supplied on real time ].

[0129] Drawing 23 is the block diagram showing the flow of a trace of log information. In drawing 23 , the client transmitting log 144 is generated by the client 300, the server receiving log 149, the output initiation termination log 151, the printing log 153, and the server transmitting log 155 are generated by the server 301, and the server receiving log 159, the output initiation termination log 161, and the printing log 163 are generated by the other servers 304. And the client transmitting log 144, the server receiving log 149, the output initiation termination log 151, the printing log 153, the server transmitting log 155, the server receiving log 159, the output initiation termination log 161, and the printing log 163 are pursued one by one with the log trace module 145.

[0130] Drawing 24 is drawing showing the example of contents of the log information stored in a log file. Although eight kinds of logs, the client transmitting log 144, the server receiving log 149, the output initiation termination log 151, the printing log 153, the server transmitting log 155, the server receiving log 159, the output initiation termination log 161, and the printing log 163, exist in drawing 22 and drawing 23 , the structure of such log information is altogether common as shown below.

[0131] The code which shows the class of log information is stored in "class of log information" 170 in drawing 24 . The code which shows a transmitting log (client) as "class of log information" 170 is stored in the client transmitting log 144. For example, to the server receiving logs 149 and 159 The code which shows a receiving log (server) as "class of log information" 170 is stored. To the output initiation

termination logs 151 and 161. The code which shows an output initiation termination log (server) as "class of log information" 170 is stored. To the printing logs 153 and 163. The code which shows a printing log (server) as "class of log information" 170 is stored, and the code which shows a transmitting log (server) as "class of log information" 170 is stored in the server transmitting log 155.

[0132] The serial number uniquely assigned on a server is stored in "identifier on server" 171. In addition, the identifier 171 on a server is a number assigned when a server receives.

[0133] The serial number uniquely assigned on the client which performed printing is stored in "identifier on client" 172. The host name of the client which performed printing is stored in "host name of client" 173.

[0134] "Output destination change names (printer etc.)" The name of the output destination change of printing is stored in 174. For example, if a printer name is stored in "output destination change name (printer etc.)" 174, printing will be performed via a server. Moreover, if the printer of other servers jurisdiction is stored in "output destination change name (printer etc.)" 174, printing which went via between servers will be performed.

[0135] "Print-out" The job title name (document name of a document or a table) 177 of the printed matter proper specified at the time of printing activation, a user name 178, a paper size 179, output number of copies 180, the output paginal order 181, the output initiation page number 182, the output termination page number 183, the output termination time 184, the output code 185, etc. are stored in 175. This "print-out" 175 is used as information for the display at the time of checking the contents of printed matter.

[0136] "Control information" A mode of operation 186, the output mode 187 which usually shows the timing of printouts, such as /time / delay, the preservation mode 188, etc. which usually show the class of printings, such as /distribution / distribution, are stored in 176. "Control information" 176 is used as information which controls activation of printing, and in case it checks under what kind of control printing has been performed, it becomes effective.

[0137] In addition, the log information between different machines can be associated by "identifier on client" 172, and the set with "the HOSU 173 TO name of a client." Thus, it becomes possible by storing different processing in the log information of the same structure to treat the I/O to the storage of a processing result in common.

[0138] Drawing 25 is a flow chart which shows the document output monitor approach concerning one example of this invention. In drawing 2525 , the log trace module 145 gains log information from the client transmitting log 144 on the client 300 which starts a log trace (step S41).

[0139] Next, the server name of the server 301 of the transmission place which transmits a spool file 142 by activation of printing is gained from the log information of the client transmitting log 144 (step S42).

[0140] Next, log information with the host name 173 and identifier 172 of the same client 300 is gained from the server receiving log 149 on the transmission place server 301 (step S43).

[0141] Next, output initiation information is acquired from the output initiation termination log 151 of a server 301 (step S44). Next, the output termination time 184 of the output initiation information acquired from the output initiation termination log 151 is set as retrieval initiation time at the time of searching log information with the host name 173 and identifier 172 of the same client 300 out of

subsequent logs (step S45).

[0142] Next, output termination information is acquired from the output initiation termination log 151 (step S46). This output termination information can be acquired only when the output is completed.

[0143] Next, if output termination information is in the log information of the output initiation termination log 151 (step S47), the output termination time of the output termination information acquired from the output initiation termination log 151 will be set as retrieval termination time (step S48).

[0144] Next, when the output destination change name 174 of the output initiation information acquired from the output initiation termination log 151 is a printer (step S49), it is the same printer (output destination change), and has the host name 173 and identifier 172 of the same client 300, and the log information which stores the output termination time 184 between retrieval initiation time and retrieval termination time is searched from the printing log 153 (step S50).

[0145] If printing to a printer from a server 301 is completed (step S51), tailing of a log will be ended by the above processing. The output destination change name 174 of the output initiation information acquired from the output initiation termination log 151 gains log information from the server transmitting log 155 on a server 301, when the spool file 147 is transmitted to the other servers 304 not from a printer but from the server 301 (step S49) (step S52). And the other servers 304 start the processing performed by the server 301, and the same processing by gaining the other servers name of the transmitted transmission place from the log information of the server transmitting log 155 (step S53), and returning to step S43.

[0146] the above processing -- printing -- completion -- or a server is changed repeatedly (step S51) and it follows until it will distinguish, if printing is not yet carried out. About the spool file 142 with same host name 173 and identifier 172 of a client, even if it specifies multiple-times printing on a server 301, the retrieval range of log information can be limited by acquiring initiation of the output initiation termination log 151, and the time of termination. Moreover, retrieval time can be shortened by limiting the retrieval range.

[0147] It relates with the information which it is as a result of different processing. Thus, by [ the ] being based on relating and following a transmitting agency It becomes possible to tell a transmitting agency about the result of processing at a transmission place. A client 300 While becoming possible to perform a printing demand, without being conscious of by which servers 301 and 304 the printer is managed, when printing is performed in which servers 301 and 304, a client 300 becomes possible [ grasping a printing situation ].

[0148] Moreover, it becomes possible about the information on processing following it to search as conditions in the time or subsequent ones by acquiring the time which the processing corresponding to one of the information ended. Moreover, in specific processing, it can check by collecting the information over initiation and termination of processing, without acquiring the information on the processing which follows [ whether the information on processing following it is completed, and ] it.

[0149] Furthermore, in case the information related from the information on the processing which follows it by acquiring initiation of processing and the time of termination is retrieved, it becomes possible to perform retrieval which specified the range.

[0150]

[Effect of the Invention] While making the drawing data generated by the drawing data generation means spool with the operating system which manages the drawing data generation means as explained above according to this invention If it is the application which operates on the operating system by storing uniquely the output-control information which controls the output of drawing data, while becoming possible to use any applications It becomes possible to add freely from after the output limiting facility with which an operating system was not equipped. The document distribution processing by the network environment which was not able to be easily realized in the conventional operating system, It becomes possible to perform notice processing of distributed printing processing, distribution place automatic detection processing or a printer error, and the completion of printing etc. to the document drawn up with general-purpose application.

[0151] Moreover, if this invention is caused like 1 voice, based on the output-control information generated by the output-control information generation means, it will become possible by generating the original printer management file corresponding to drawing data to perform processing and actuation of a printer management file in the original format which became independent of an operating system. For example, when Windows is used as an operating system, it becomes possible as a communication network between a client and a server to use various communication networks, such as LAN, WAN, and the Internet, by changing a printer management file into an original format to being limited to LAN etc.

[0152] Moreover, if this invention is caused like 1 voice, it is based on the retrieval conditions sent from the client. By comparing with the distribution place name in which the character string which is contained in a document, and which was specified beforehand is detected automatically, and the detected character string is stored by the distribution place definition table While becoming possible from a print job to detect a distribution place automatically It becomes possible to print automatically to all the distribution places that correspond to retrieval conditions only by performing one printing directions, without becoming possible to specify an output unit based on the distribution place detected automatically, and specifying a distribution place.

[0153] If this invention is caused like 1 voice, the rectangle field which divided the imagination form moreover, by making assignment possible as a distribution place candidate's code area It becomes possible to specify a code area in the direction of the four directions of a form, middle, etc., and it becomes possible to set up the code area of a distribution place appropriately by one field setup in an imagination form, when a paper size differs from how to place etc. in each printing document.

[0154] Moreover, only by specifying one distribution place by registering the correspondence relation between the distribution place name for N base, and an output unit, or defining two or more output units as one logical distribution place, if this invention is caused like 1 voice While becoming possible to determine automatically two or more output units corresponding to the distribution place and becoming possible to specify a distribution place not by the physical output unit but by the logical name, it becomes possible to distribute a document to two or more output units by one assignment. Consequently, it becomes that it is possible in printing automatically to all the distribution places that correspond to conditions with one printing directions, without becoming possible to save the time and effort which makes the multiple selection of the distribution place to wish at the time of printing activation, specifies a distribution place for every printing document, or specifies all the names of the printer which is a

candidate for distribution, and being conscious of the printer name corresponding to a distribution place. [0155] If this invention is caused like 1 voice, moreover, by storing the transmitting result, the receiving result, output, and printing result of a spool file with printing demand former information [ when a transfer of the spool file on a network is performed ] The transmitting result, the receiving result, output, and printing result of the spool file become possible [ judging easily ] about whether for it to have originated in which printing demand origin, and to have been obtained, and it becomes possible to recognize how the printing demand of it was processed by the printer of which server. Consequently, when printing is performed in which server, a client becomes possible [ grasping a printing situation correctly ], while becoming possible [ performing a printing demand ], without being conscious of by which server the printer is managed.

---

## DESCRIPTION OF DRAWINGS

---

### [Brief Description of the Drawings]

[Drawing 1] It is the block diagram showing the printing structure of a system concerning one example of this invention.

[Drawing 2] It is a block diagram explaining the configuration of the client control section of drawing 1 .

[Drawing 3] It is a block diagram explaining the configuration of the server control section of drawing 1 .

[Drawing 4] It is a block diagram explaining the configuration of the printer output-control section of drawing 1 .

[Drawing 5] It is a block diagram explaining actuation of a virtual driver.

[Drawing 6] It is drawing showing the example of contents of a job description file.

[Drawing 7] It is drawing showing the example of contents of an output-control file.

[Drawing 8] It is the flow chart which shows actuation of a virtual driver.

[Drawing 9] It is drawing showing the setting screen of DEVMODE information.

[Drawing 10] It is drawing showing the setting screen of automatic distribution information.

[Drawing 11] It is drawing explaining the specification method of the code area concerning one example of this invention.

[Drawing 12] It is drawing showing the setting screen of assignment distribution information.

[Drawing 13] It is drawing showing the setting screen of the distributed printed information concerning one example of this invention.

[Drawing 14] It is the block diagram showing the configuration of a CAPCAL print processor.

[Drawing 15] It is drawing showing the example of contents of a CAPCAL job description file.

[Drawing 16] It is the flow chart which shows actuation of a CAPCAL print processor.

[Drawing 17] It is a block diagram explaining the automatic distribution processing concerning one example of this invention.

[Drawing 18] It is drawing showing the example of a retrieval condition at the time of automatic distribution.

[Drawing 19] It is the flow chart which shows actuation of the automatic distribution processing concerning one example of this invention.

[Drawing 20] It is a block diagram explaining the assignment distribution processing concerning one example of this invention.

[Drawing 21] It is the flow chart which shows actuation of the assignment distribution processing concerning one example of this invention.

[Drawing 22] It is the block diagram showing the configuration of the document output monitor processing section concerning one example of this invention.

[Drawing 23] It is the block diagram showing the flow of a trace of log information.

[Drawing 24] It is drawing showing the example of contents of log information.

[Drawing 25] It is the flow chart which shows the document output monitor approach concerning one example of this invention.

[Drawing 26] It is a block diagram explaining the conventional printing approach.

[Drawing 27] It is a block diagram explaining the conventional distributed printing approach.

**[Description of Notations]**

300 Client

301 304 Server

302, 303, 28, 29, 44 Communication line

1 Application

2 Seven GDI

3 Five Windows printing control section

4 Client Control Section

6 Server Control Section

8 Printer Output-Control Section

9 Printer

11 Spooler Process

12 Provider

13 47 Print processor

14 Virtual Driver

15 CAPCAL Driver Property Control Thread

16 System Spool Queue

17 Output-Control File

18 Job Description File

19, 24, 35, 40 EMF file

20 CAPCAL Client Definition File

21 CAPCAL Print Processor

22 CAPCAL Spool Queue

23, 34, 39 CAPCAL job description file

25 CAPCAL Destination Definition File

26 CAPCAL Client Main Process

27 CAPCAL Job Transfer Thread

30 31 CAPCAL receiving thread

32 CAPCAL Server Main Process

- 33 CAPCAL Spool Document Queue
- 36 CAPCAL Server Side Definition File
- 37 Distribution Distributed-Processing Thread
- 38 CAPCAL Spool Printer Queue
- 41 42 Output thread classified by printer
- 43 CAPCAL Job Transfer Thread
- 45 Printer Driver
- 46 Raw Spool File
- 48 51 SENTORO port monitor
- 49 Spooler Process Main
- 50 Printer Output Management Department
- 52 53 Screen call
- 54 Spool System
- 55 Port Thread
- 57,142,147,157 Spool file
- 61 Job Description File Header
- 62 DEVMODE Information
- 63 Job Description File Footer
- 64 Printer Driver Private Information
- 71 CAPCAL Output-Control File Header
- 72 Mode-of-Operation Information
- 73 Document Additional Information
- 74 Document Attribute Information
- 75 Output-Control Information
- 81 User Interface
- 82 Block-Definition Control Section
- 83 85 89 91 Form
- 84, 86, 88, 90, 92 The appointed field
- 87 Virtual Form
- 101 Print-Data Reading Control Section
- 102 Print-Data Storing Section
- 103 Print-Data Configuration Section
- 104 CAPCAL Spool File Generation Section
- 105 Page Management Information Storing Section
- 106 Print Job Information Reading Control Section
- 107 CAPCAL Output-Control File Reading Control Section
- 108 CAPCAL User Information Reading Control Section
- 109 CAPCAL Job Information Storing Section
- 110 Print Job Information Configuration Section
- 111 CAPCAL Job Description File Header
- 112 Windows Job Information

- 113 DEVMODE Information
- 114 Output-Control Information
- 115 CAPCAL User Information
- 116 Print-Data Page Management Information
- 121 Print Job
- 122 Character String
- 123 132 Registry
- 124 Retrieval Conditions
- 125 Comparator
- 131 Distribution Place Information
- 143 154 Transfer module
- 144 Client Transmitting Log
- 145 Log Trace Module
- 148 158 Receiving module
- 149 159 Server receiving log
- 150 160 Output-control module
- 151 161 Output initiation termination log
- 152 162 Printing control module
- 153 163 Printing log
- 155 Server Transmitting Log
- 170 Class of Log Information
- 171 Identifier on Server
- 172 Identifier on Client
- 173 Host Name of Client
- 174 Output Destination Change Names (Printer Etc.)
- 175 Print-out
- 176 Control Information
- 177 Job Title Name
- 178 User Name
- 179 Paper Size
- 180 Output Number of Copies
- 181 Output Paginal Order
- 182 Output Initiation Page Number
- 183 Output Termination Page Number
- 184 Output Termination Time
- 185 Output Code
- 186 Mode of Operation
- 187 Output Mode
- 188 Preservation Mode

---

[Translation done.]

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**